

AD-A136 253

SURVEILLANCE AVOIDANCE TECHNIQUE DEMONSTRATION(U)  
ADVANCED INFORMATION AND DECISION SYSTEMS MOUNTAIN VIEW  
CA J M ABRAM ET AL. 02 DEC 83 A1/OS-TR-1026/10-1

1/2

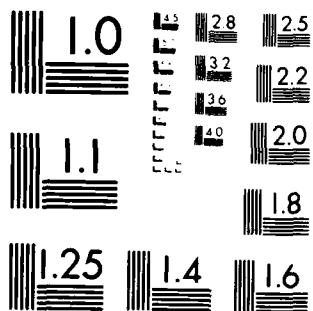
UNCLASSIFIED

N00014-82-C-0085

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

# AIDS

— TR-1026/10-1  
November 2, 1983

A136253

## SURVEILLANCE AVOIDANCE TECHNIQUE DEMONSTRATION

### FINAL REPORT

U.S. Government Contract No. N00014-82-C-0085

#### Prepared for:

Naval Electronic Systems Command  
PME 120-113  
Arlington, VA 20360  
and  
Office of Naval Research  
Arlington, VA 22217

#### Prepared by:

J.M. Abram  
J.R. Payne  
E.A. Domeshek

DTIC  
S E  
DEC 27 1983  
H

Approved for public release; distribution unlimited.

ADVANCED INFORMATION & DECISION SYSTEMS

Mountain View, CA 94040

83 12 27 11

DMC FILE COPY

Surveillance Avoidance  
TABLE OF CONTENTS

Final Report

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1. INTRODUCTION. . . . .	1-1
1.1 OBJECTIVES . . . . .	1-1
1.2 BACKGROUND . . . . .	1-2
1.3 SYSTEM SUMMARY . . . . .	1-4
1.4 REPORT ORGANIZATION . . . . .	1-6
2. SYSTEM HARDWARE . . . . .	2-1
3. SYSTEM SOFTWARE . . . . .	3-1
3.1 APPLICATIONS PROGRAM . . . . .	3-1
3.1.1 Top-Level Executive Software . . . . .	3-1
3.1.2 SELECT MAP Function . . . . .	3-3
3.1.3 SPECIFICATION Function . . . . .	3-6
3.1.4 SENSOR SCENARIO Function . . . . .	3-25
3.1.5 OPTIMIZATION Function . . . . .	3-47
3.1.6 CREATE PATH Function . . . . .	3-56
3.1.7 SHOW RESULTS Function . . . . .	3-56
3.2 MAN-MACHINE INTERFACE SOFTWARE . . . . .	3-66
3.2.1 User Interaction with the Interface . . . . .	3-68
3.2.2 Implementation of the Man-Machine Interface . . . . .	3-71
4. A SURVEILLANCE AVOIDANCE EXAMPLE . . . . .	4-1
4.1 INTRODUCTION TO ISAS . . . . .	4-1
4.2 PROBLEM SPECIFICATION . . . . .	4-1
4.3 PATH OPTIMIZATION . . . . .	4-15
4.4 EXAMINING THE RESULTS . . . . .	4-15
5. ENGINEERING ASSESSMENT . . . . .	5-1
5.1 CURRENT MEMORY AND RUNTIME REQUIREMENTS . . . . .	5-1
5.1.1 Memory Requirements . . . . .	5-1
5.1.2 Runtime Requirements . . . . .	5-2
5.2 ENGINEERING ASSESSMENT FOR FUTURE IMPLEMENTATION . . . . .	5-3
5.2.1 Assessment of Memory Needs . . . . .	5-3
5.2.2 Assessment of Runtime Needs . . . . .	5-4
6. CONCLUSIONS . . . . .	6-1
6.1 POTENTIAL FUTURE EFFORTS . . . . .	6-1
6.2 SUMMARY OF WORK PERFORMED . . . . .	6-4



100	
Availability Codes	
Distribution/for	
Dist	Control
A-1	

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
2-1	Demonstration Hardware Configuration . . . . .	2-2
3-1	Applications Program Structure . . . . .	3-2
3-2	SELECT MAP Menu . . . . .	3-4
3-3	Summary of SELECT MAP Function . . . . .	3-5
3-4	Indian Ocean Map . . . . .	3-7
3-5	SPECIFICATION Menu . . . . .	3-8
3-6	Summary of SPEC Function . . . . .	3-9
3-7	Summary of BASIC Module . . . . .	3-11
3-8	Data Form for BASIC . . . . .	3-12
3-9	Data Form for SET QUANTIZATION . . . . .	3-13
3-10	Resolution Options . . . . .	3-16
3-11	Summary of SET QUANTIZATION Module . . . . .	3-17
3-12	Summary of RENDEZVOUS Module . . . . .	3-18
3-13	Data Form for RENDEZVOUS . . . . .	3-19
3-14	Summary of EXCLUSION Module . . . . .	3-21
3-15	Data Form for EXCLUSION . . . . .	3-22
3-16	Summary of CHECK AND QUANTIZE Module . . . . .	3-23
3-17	Construction of Problem Grid . . . . .	3-24
3-18	Summary of SENSOR SCENARIO Function . . . . .	3-26
3-19	Menu for PLATFORM LIST . . . . .	3-28
3-20	Summary of PLATFORM LIST Module . . . . .	3-29
3-21	Data Form for EARTH-BASED, FIXED PLATFORM . . . . .	3-30
3-22	Data Form for EARTH-BASED, BARRIER SEARCH PLATFORM . . . . .	3-31
3-23	Data Form for EARTH-BASED, TRANSITING PLATFORM . . . . .	3-32
3-24	Data Form for SATELLITE PLATFORM . . . . .	3-33
3-25	Summary of EDIT PLATFORM Module . . . . .	3-34
3-26	Summary of SEE SENSOR Module . . . . .	3-36
3-27	Sensor Model for SOSUS . . . . .	3-37
3-28	Sensor Model for ESM . . . . .	3-38
3-29	Sensor Model for OTH . . . . .	3-39
3-30	Sensor Model for Search Radar . . . . .	3-40
3-31	Sensor Model for TASS. . . . .	3-41
3-32	Sensor Model for Satellite ROR . . . . .	3-42
3-33	Sensor Model for Satellite IR. . . . .	3-43
3-34	Summary of COMPUTE PROBABILITIES Module . . . . .	3-44
3-34	Summary of COMPUTE PROBABILITIES Module (Cont.). . . . .	3-45
3-34	Summary of COMPUTE PROBABILITIES Module (Cont.). . . . .	3-46
3-35	Summary of OPTIMIZATION Function . . . . .	3-48
3-36	Summary of DP Module . . . . .	3-49
3-36	Summary of DP Module (Cont.) . . . . .	3-50
3-36	Summary of DP Module (Cont.) . . . . .	3-51

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
3-37	Summary of EVALUATION Module . . . . .	3-53
3-38	Initial OPTIMIZATION Form. . . . .	3-54
3-39	Name Path Form . . . . .	3-55
3-40	OPTIMIZATION Ended Successfully Form . . . . .	3-57
3-41	OPTIMIZATION Failed Form . . . . .	3-58
3-42	Summary of SHOW RESULTS Function . . . . .	3-59
3-43	Menu for SHOW RESULTS. . . . .	3-60
3-44	GRAPHICAL Path Menu. . . . .	3-62
3-45	Sample Graphical Display . . . . .	3-63
3-46	Textual Display for GRAPHICAL Module . . . . .	3-64
3-47	Summary of GRAPHICAL Module. . . . .	3-65
3-48	Example of Tabular Summary . . . . .	3-67
3-49	Help Message for QUANTIZATION. . . . .	3-69
4-1	ISAS Title Page. . . . .	4-2
4-2	Page 1 of Tutorial . . . . .	4-3
4-3	Page 2 of Tutorial . . . . .	4-4
4-4	Page 1 of General Instructions . . . . .	4-5
4-5	Page 2 of General Instructions . . . . .	4-6
4-6	Page 3 of General Instructions . . . . .	4-7
4-7	Page 4 of General Instructions . . . . .	4-8
4-8	Completed Describe Basic Problem Form. . . . .	4-10
4-9	Completed Set Quantization Form. . . . .	4-11
4-10	Completed Enter Exclusion Areas Form . . . . .	4-12
4-11	List of Platforms for Example. . . . .	4-13
4-12	Pictorial Summary of Sensor Scenario . . . . .	4-14
4-13	OPTPATH Menu . . . . .	4-16
4-14	Path 1 Brief Summary . . . . .	4-17
4-15	Path 2 Brief Summary . . . . .	4-18
4-16	Path 3 Brief Summary . . . . .	4-19
4-17	Path 4 Brief Summary . . . . .	4-20
4-18	Path 5 Brief Summary . . . . .	4-21
4-19	Path 6 Brief Summary . . . . .	4-22
4-20	End of Optimization for Example. . . . .	4-23
4-21	SHOWPATH Menu. . . . .	4-24
4-22	TABULAR Path Menu. . . . .	4-26
4-23	Graphical Display for Stage 2. . . . .	4-27
4-24	Tabular Information for Stage 2. . . . .	4-28
4-25	Graphical Display for Stage 9. . . . .	4-30
4-26	Tabular Information for Stage 9. . . . .	4-31
4-27	Graphical Display for Stage 11 . . . . .	4-32

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
4-28	Tabular Information for Stage 11 . . . . .	4-33
4-29	Graphical Display for Stage 17 . . . . .	4-34
4-30	Tabular Information for Stage 17 . . . . .	4-35
4-31	Graphical Display for Stage 31 . . . . .	4-36
4-32	Tabular Information for Stage 31 . . . . .	4-37
4-33	Respecify Menu After SHOW RESULTS. . . . .	4-39

## 1. INTRODUCTION

This final report summarizes the work performed by Advanced Information & Decision Systems in developing an Interactive Surveillance Avoidance System (ISAS), in accordance with the requirements of paragraph a.(2) in the Task Description Statement of Contract N00014-82-C-0085, Modification P00001. This document includes descriptions of the hardware, software, and man-machine interfaces used in ISAS, an example of a typical session of using the system, and an assessment of memory and runtime requirements for both the current implementation and for a proposed small computer implementation.

### 1.1 OBJECTIVES

The objectives of ~~this~~ (ISAS) development effort were to: 1) demonstrate a process for generating near-optimal ship transit routes that avoid surveillance by enemy sensors located in space, in the air, on the surface, or undersea; and 2) demonstrate a friendly computer environment in which the user can easily provide necessary information to the system and in which the system displays information in an easily comprehended form to the user. The analytical procedure for generating an optimal path was demonstrated, as was a man-machine interface that fosters interaction between the transit planner and the system.

This project was to demonstrate the feasibility of the concept and to estimate the computer system requirements for introducing the capability into the fleet environment. ISAS could be useful to the fleet in planning transits before leaving port, and in re-planning the remainder of a transit when new information becomes available about enemy sensors, weather forecasts, or changes in final destination points, rendezvous points and times.

ISAS will be an analytical aid important to deploying our forces with long range weapons (aircraft carriers and Tomahawk capable ships



and submarines) to their areas of operation. ISAS will exploit weaknesses or gaps in the enemy's surveillance coverage, making it more difficult for him to detect our forces and then target them with his own long range weapons.

## 1.2 BACKGROUND

There has been a great deal of analytical work done to support the planning of tactical surveillance missions. Much of this work is based on B.O. Koopman's OEG-56 report on surveillance and has been encoded into tactical search plans for the various types of surveillance platforms. As new sensors and surveillance platforms (from fixed undersea acoustic sensors to satellite borne systems) have been developed, new tactical search plans have also been developed that exploit the new capabilities. However, comparatively little effort has been directed toward finding the best tactics for defeating a set of surveillance systems, especially in preventing them from detecting a target of interest.

Some at-sea experiments have been conducted to explore defeating special classes of sensor systems. For example, the Third Fleet UPTIDE exercises explored several techniques to prevent submarines from, or at least delay them in, detecting (acoustically) and identifying a carrier. Also there have been a few extended transits in strict EMCON to explore this method of defeating HF/DF and other ESM systems. However, in most exercises the need for training of all types, meeting replenishment groups, and limited time at sea, have tended to preclude practicing or evaluating detection avoidance techniques against all types of enemy sensors.

The introduction of long range missiles (and more recently long range supersonic aircraft) into the military inventories of our potential adversaries has provided them with a capability to relatively quickly strike our naval forces at large distances from areas they may control through firepower superiority. Hence it has become increasingly important for us to find ways of transiting our own ships and their long range weapon systems into the destination areas required for successful

mission accomplishment, without their being attacked enroute. Minimizing the chance of detection by all the enemy's sensors is an important step that should be taken to thwart the enemy's ability to attack our forces in transit.

Several analytical aids have been developed that can aid a planner, but most of these have taken the form of presenting where enemy sensors are located, together with their fields of coverage. Satellite coverage intervals may be given. It is then left to the planner to decide what path and speed to take through the enemy sensor field. For planning purposes "cookie cutter" models of the sensors are often used, although detailed sensor performance model outputs can be requested from the Fleet Numerical Oceanographic Center for specific sensor types, targets, and environmental conditions.

One analytical model, SURVAV, was developed by Decisions and Designs, Inc. (DDI) in 1977 as a decision aid for planning transits that avoid detection by satellites. This model was demonstrated in the Advanced Command and Control Architectural Testbed (ACCAT) at the Naval Ocean Systems Center (NOSC). Several Fleet Admirals and staff have told the NOSC staff responsible for the development of the Tactical Flag Command Center (TFCC), colocated with the ACCAT, that a surveillance avoidance planning aid is needed in the Fleet. But to date SURVAV has found use primarily in classes and research in the ACCAT's remote module at the U.S. Naval Postgraduate School.

AI&DS personnel observed SURVAV in the ACCAT and noted that it relied on the user to specify the path (resulting in a sub-optimal path), did not treat fuel as a constraint, considered only satellite sensors, and was slow. The surveillance avoidance problem is analogous to the aircraft penetration of air defenses problem for which AI&DS has developed an analytical procedure that relies on dynamic programming for finding an optimal trajectory within the fuel constraint of the problem.

Interest was expressed in AI&DS' approach by personnel in TFCC, CINCPACFLT TAC D&E, NAVELEX 612, ONR and NAVELEX PME-120. The current ISAS development is sponsored by PME-120 through ONR. ISAS, which is free of most of the limitations mentioned above, has been developed has been demonstrated on general purpose AI&DS hardware, including a VAX 11/750 and a color graphics monitor. However this implementation at AI&DS takes into account the subsequent goal of implementing the system on a smaller Navy computer.

### 1.3 SYSTEM SUMMARY

ISAS consists of both analytical and man-machine interface software implemented on a variety of hardware items. The hardware consists of a VAX 11/750 minicomputer, an Ann Arbor Ambassador CRT terminal, a color graphics monitor, a CROMEMCO microprocessor that serves as interface between the VAX and the graphics monitor, and a line printer.

The user interacts with ISAS through the Ann Arbor Ambassador terminal. The inputs required by ISAS are accomplished through the use of approximately thirty five selection menus and data entry forms. On each of these forms, cursor control keys are used to move to menu items or data input fields and other special keys are used for functions such as enter, remove, add, and help. A combination of control software and user control guides the user through the ISAS system until a transit problem is specified and solved.

The software is written in a highly modular fashion. This modularity will facilitate further development of and improvements to the code such as replacing or adding sensor and sensor platform motion models, modifying cost functions, or other software modifications. Modularity will also facilitate implementation of the software on a Navy computer. For instance, a module to read sensor platform motion models from a file on the VAX could be replaced by a module that provides an interface from a Navy data base system to the surveillance avoidance system.

The mathematical technique used in automatically generating paths is dynamic programming (DP). The DP approach was chosen over techniques such as nonlinear programming because the surveillance avoidance problem can be formulated as a DP problem without requiring major approximations, and DP always finds global rather than local extremum points.

DP is used for solving problems that require multiple decisions. Decision points are referred to as stages. At any decision point there are a number of states that characterize the current situation. Different states in a given stage can be reached by different sequences of decisions from the initial state. At each state in a given stage, a number of decisions are possible, each decision causing a transition to some state in the subsequent stage at some cost or reward. The DP technique searches for the decision sequence that leads from the initial state to the final state at minimum overall cost or maximum overall reward. A more detailed description of dynamic programming and of the specific formulation of the surveillance avoidance problem as a DP program can be found in Appendix A.

The cost function to be minimized is overall probability of detection. The fuel constraint is treated by incorporating fuel usage into the probability of detection cost function via a Lagrange multiplier. This multiplier determines the amount of weight given to fuel usage, as opposed to detection probability, in the cost function. A solution using  $F$  units of fuel will be globally optimal over all paths using no more than  $F$  fuel units. If  $F$  is less than the maximum allowable fuel usage, then a better path using more than  $F$  units of fuel may exist. In most cases, it is expected that the actual optimal solution will be found by the DP algorithm, but occasionally the solution will only be close to optimal. In the remainder of this document, the word "optimal" will be used rather than "near-optimal".

#### 1.4 REPORT ORGANIZATION

The hardware configuration for the system is described in Section 2. Section 3 contains a description of the software for ISAS, including executive, analytical, and man-machine interface software. Section 4 steps through a sample ISAS session. In Section 5, memory and runtime requirements for the current implementation and an engineering assessment for a smaller computer are given. Conclusions and directions for further development are given in Section 6.

## 2. SYSTEM HARDWARE

The demonstration version of ISAS was developed using in-house hardware. A VAX 11/750 provides the bulk of the computer power. This is a 32 bit minicomputer with a 300 Mbyte disk. The VAX performs all numerical computations, controls the flow of operations, runs the man-machine interface software, and provides all of the memory required by ISAS.

The primary user input/output device is an Ann Arbor Ambassador CRT terminal. In addition to a standard alphanumeric keyboard, the Ann Arbor has several programmable special function keys. These special keys are used for cursor control and menu selection. All user inputs enter the system through the Ann Arbor, and alphanumeric outputs, such as menus, data forms, and data summarizing a path of interest, are displayed on the Ann Arbor screen.

A Hitachi HM-2719 color graphics monitor is used to display maps, routes, sensor coverages, and other information of a graphical nature. The color monitor is used because of in-house hardware capabilities and because certain types of displays are more easily understandable with the use of several colors. However, any display of information used in ISAS could be presented in a format suitable for black-and-white graphics monitors. A CROMEMCO System 3 microprocessor serves as an interface between the VAX and the graphics monitor. Some hard copy outputs are provided by a Printronix 600 line matrix printer. The hardware configuration is shown in Figure 2-1.

In-house hardware was used to demonstrate the feasibility of the dynamic programming technique for surveillance avoidance. An operational version of ISAS would be required to run on existing Navy hardware.

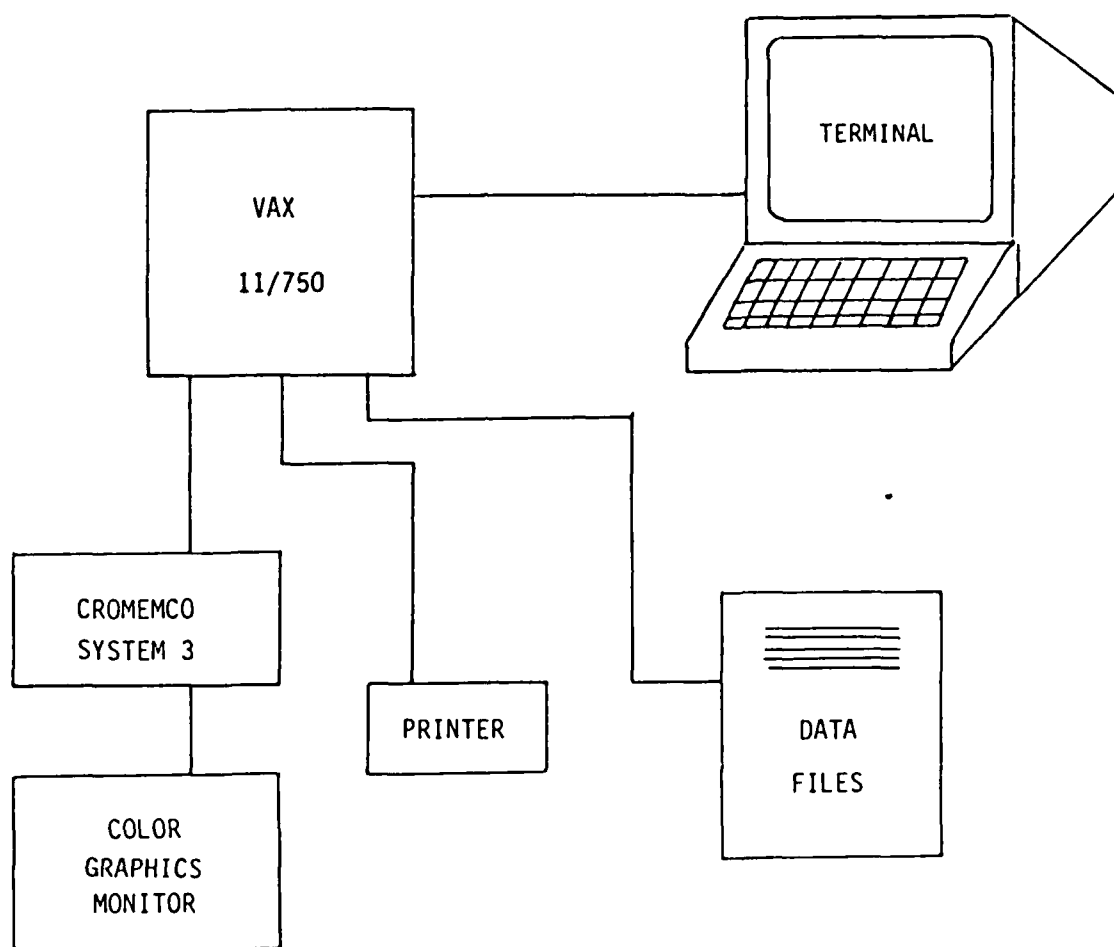


Figure 2-1: Demonstration Hardware Configuration

### 3. SYSTEM SOFTWARE

The ISAS software consists of two major components: the applications program and the man-machine interface. The applications program is written in FORTRAN and contains all of the executive and analytical subroutines required by ISAS to solve the surveillance avoidance problem. The man-machine interface consists of a number of generic subroutines written in C and LISP and an experimental relational data base system called TROLL. The interface software is not ISAS-specific, but the information describing the forms and menus used by ISAS must be pre-stored into the data base. In this section, the two ISAS software components are described.

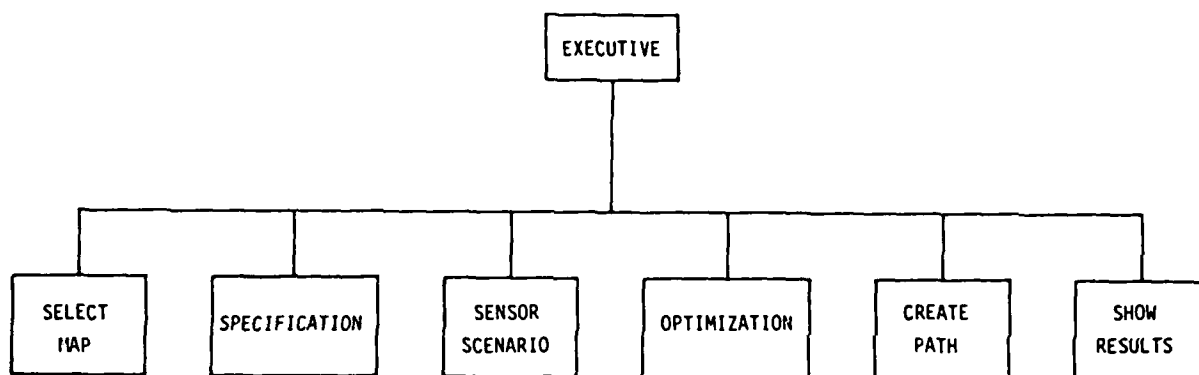
#### 3.1 APPLICATIONS PROGRAM

The applications program has a top-level executive, six functional modules (each with its own executive software), and a number of analytical subroutines that are called as needed from within the six functions. The structure of the applications program can be seen in Fig. 3-1. The applications program also contains calls to the man-machine interface (MMI) buffer routines, as will be discussed in the section on the interface. In this section, each component of the applications software is described.

##### 3.1.1 Top-Level Executive Software

The executive software in ISAS is responsible for controlling the flow of operations. Executive software exists at more than one level within the system. At the highest level, the executive controls the order in which the user is allowed to proceed through the ISAS functions. When the user has completed one of the functions and wishes to proceed to the next function, there are two possible situations: 1) the user has no choice as to the function to be performed next and the executive automatically takes him to the next function, or 2) the user is





EXECUTIVE	EXECUTIVE SOFTWARE FOR SYSTEM CONTROL
SELECT MAP	CHOOSE DESIRED MAP
SPECIFICATION	INTERACTIVE PROBLEM SPECIFICATION
SENSOR SCENARIO	EXAMINE/EDIT SENSOR SCENARIO
OPTIMIZATION	COMPUTATION OF OPTIMAL PATH
CREATE PATH*	INTERACTIVE PATH CREATION
SHOW RESULTS	DISPLAY CERTAIN EVALUATION RESULTS

\*CREATE PATH WAS NOT IMPLEMENTED IN THE DEMONSTRATION VERSION OF ISAS

Figure 3-1: Applications Program Structure

presented with a menu of admissible choices, the man-machine interface passes the choice to the executive, and the executive calls the requested function. The executive prevents the user from attempting to execute a function whose prerequisite functions have not been completed. The user also has the option of regressing to a previously executed function. Again, the MMI tells the executive which function the user has requested so that the appropriate action can be taken.

The top-level executive can also supply various information to the user. Instructions on the use of the MMI are provided by the executive at the beginning of each ISAS session. Those who do not wish to read this basic tutorial can skip over it. General instructions describing the basic ISAS functions and the order in which they are to be performed are also available to the user on request. The executive is also designed to provide specific help messages within any part of ISAS when requested. The MMI tells the executive which part of which function the user is working on, and the executive then puts up the appropriate information. At the present, only a few of the help messages have been written, but the structure exists for adding the remainder.

The means to exit ISAS are also provided by the top-level executive. Exiting ISAS causes all user entered inputs to be erased so that a new session can be started from the beginning. As a safety precaution, the user must verify that he wants to exit ISAS before the session is terminated.

### 3.1.2 SELECT MAP Function

In this function, the user selects a map from a menu of available options, such as the one in Fig. 3-2. The user should pick that map that most closely fits his needs with respect to the region of the world and map scale of interest for his particular transit problem. After the user selects a map, an image that contains land masses, water regions, and latitude/longitude lines for that map is displayed on the graphics monitor. The user may examine as many maps as he desires until the most suitable map is found. Figure 3-3 gives a functional description of

- General Instructions
  - Exit ISAS
  - Proceed to Next Task
- 

CHOOSE A MAP  
(Position Cursor: Press "ENTER")

- Indian Ocean (North)
- Arabian Sea

Figure 3-2: SELECT MAP Menu

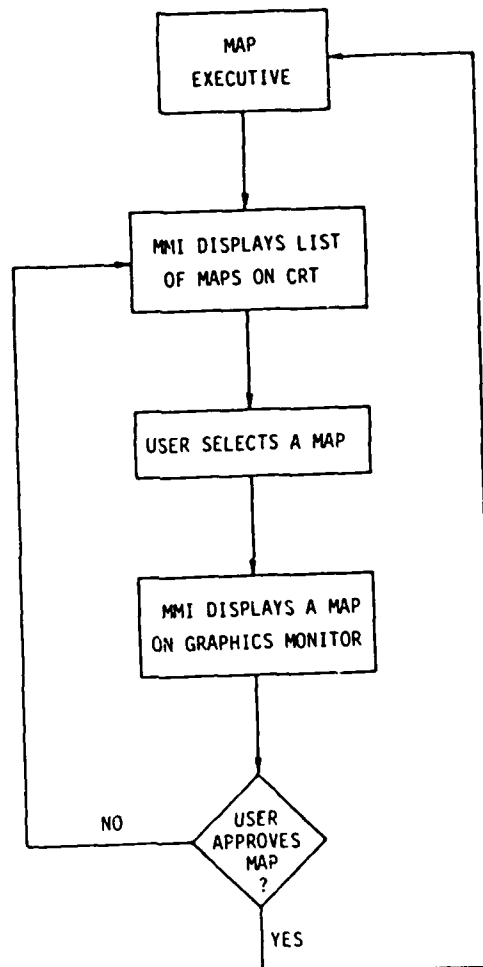


Figure 3-3: Summary of SELECT MAP Function

#### SELECT MAP.

The data required by SELECT MAP is stored in the form of bit maps that represent land, water, or latitude/longitude line. The demonstration version of ISAS has only a map of the northern part of the Indian Ocean, shown in Fig. 3-4. An operational version of ISAS would interface to an extensive map data base. An interactive system where the user selects an area of the world and then enlarges or reduces the map to get the best scaling factor would be a good approach to map selection. The exact procedure used for map selection in an operational version would, of course, depend on the type of map data base available.

#### 3.1.3 SPECIFICATION Function

In the SPECIFICATION function the user is requested to supply all of the information that describes the transit requirements. The user is first presented a menu of the four SPECIFICATION modules, as seen in Fig. 3-5. BASIC and SET QUANTIZATION must be performed before the user is allowed to proceed to the next task; RENDEZ VOUS and EXCLUSION are optional tasks. The user is free to enter these modules in any order and as many times as he wishes until he is ready to proceed. Upon exiting any module, constraints for the inputs associated with that module are checked by the SPECIFICATION executive. If there are no constraint violations, the user is allowed to exit. If there is a violation, the user is taken back to the module, the cursor is positioned at the erroneous input box, and an explanation of the violation is posted at the bottom of the form. The user must either correct the error or signify that he wishes to defer the error before he can leave the module. In addition to these four interactive modules, SPECIFICATION has an analytical module, CHECK AND QUANTIZE, that is executed just before proceeding to the next function. Figure 3-6 shows the structure of SPECIFICATION. Each of the five modules is now described.

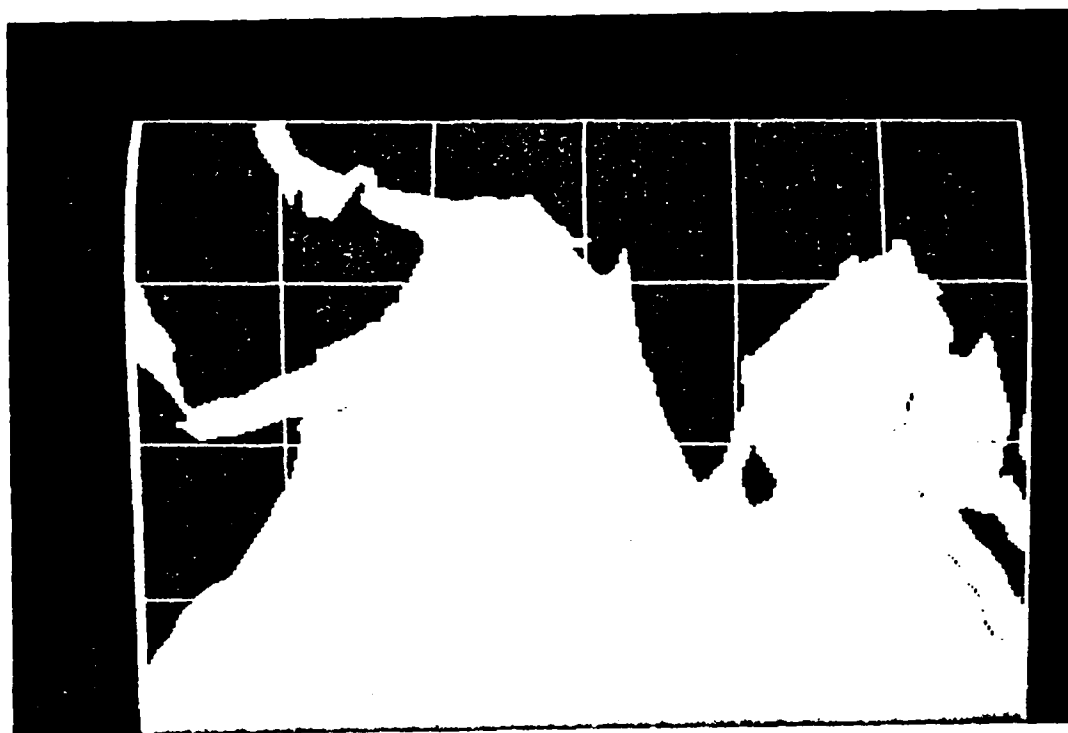


Figure 3-4: Indian Ocean Map

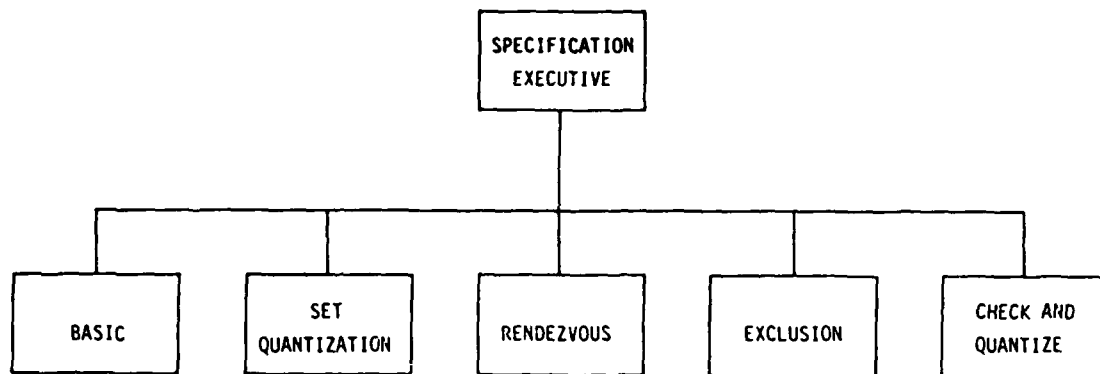
- General Instructions	- Exit ISAS
- Re-Specify Problem	- Proceed to Next Task

---

SPECIFY A TRANSIT PROBLEM  
(Position Cursor. Press "ENTER")

- Describe Basic Problem (Required)	- Set Quantisation (Required)
- Enter Rendezvous Point (Optional)	- Enter Exclusion Areas (Optional)

Figure 3-5: SPECIFICATION Menu



BASIC	INPUT BASIC TRANSIT DESCRIPTION
SET QUANTIZATION	INPUT QUANTIZATION PARAMETERS
RENDEZVOUS	INPUT RENDEZVOUS POINT
EXCLUSION	INPUT EXCLUSION AREAS
CHECK AND QUANTIZE	FINAL CONSTRAINT CHECKS AND QUANTIZATION CALCULATIONS

Figure 3-6: Summary of SPEC Function



### 3.1.3.1 BASIC Module

In BASIC (see Fig. 3-7), the user is required to provide the information requested on the data form shown in Fig. 3-8. By moving the cursor from data field to data field the user inputs the start and end positions, dates, and times for the transit, the minimum speed, maximum speed, and maximum allowable fuel in units of percentage of ship capacity for the transit, and selects his ship type from the menu at the bottom of the form. (Ship type is needed to find the fuel consumption model, fuel capacity, and physical maximum speed of the transiting ship for subsequent calculations and constraint checks.)

There are a number of constraint checks associated with BASIC. Latitude must be between 0 and 90 degrees north or south and longitude must be between 0 and 180 degrees east or west. The start and finish positions must be on the map and in the water. The dates and times must be valid (month is between 1 and 12, if month is 1 then day is between 1 and 31, etc.) and the finish date/time must be later than the start date/time. Minimum and maximum speed must be nonnegative, and maximum speed must be no smaller than minimum speed and no greater than the physical capability of the appropriate ship type. Maximum allowed fuel consumption must be between 0 and 100 percent of capacity. Finally, it must be feasible to traverse the great circle path between start and finish in the allowed time without exceeding the maximum speed or maximum allowed fuel consumption.

### 3.1.3.2 SET QUANTIZATION Module

The dynamic programming algorithm in ISAS requires a quantization of space into a grid with a finite number of cells and of time into a finite number of uniform intervals. The size of the cells and the length of each interval are variables that depend on the problem at hand and on the wishes of the user. In SET QUANTIZATION the user is presented with the form depicted in Fig. 3-9 and is asked to provide guidance on the choice of quantization parameters.

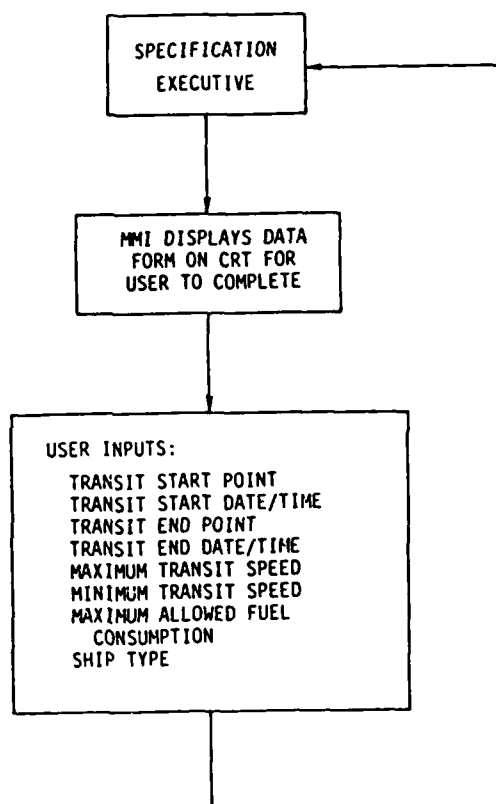


Figure 3-7: Summary of BASIC Module

<input type="checkbox"/> General Instructions	<input type="checkbox"/> Exit ISAS
<input type="checkbox"/> Re-Specify Problem	<input type="checkbox"/> Proceed to Next Task

---

SPECIFY A TRANSIT PROBLEM  
(Position Cursor: Press "ENTER")

<input checked="" type="checkbox"/> Describe Basic Problem (Required)	<input type="checkbox"/> Set Quantisation (Required)
<input type="checkbox"/> Enter Rendezvous Point (Optional)	<input type="checkbox"/> Enter Exclusion Areas (Optional)

---

Describe Basic Problem (Required)  
(Position Cursor; Enter Data)

Transit	Latitude	Longitude	Date	Time(GMT)
Start			/ /	
Finish			/ /	

Describe Your Ship -

Maximum Speed (knots)	:
Minimum Speed (knots)	:
Maximum Allowed Fuel Consumption (% Fuel Capacity)	:

Ship Type (Position Cursor: Press ENTER)

<input type="checkbox"/> Cruiser
<input type="checkbox"/> Destroyer
<input type="checkbox"/> Aircraft Carrier

Figure 3-8: Data Form for BASIC

<input type="checkbox"/> General Instructions	<input type="checkbox"/> Exit ISAS
<input type="checkbox"/> Re-Specify Problem	<input type="checkbox"/> Proceed to Next Task
<hr/>	
SPECIFY A TRANSIT PROBLEM	
(Position Cursor: Press "ENTER")	
<input type="checkbox"/> Describe Basic Problem (Required)	<input checked="" type="checkbox"/> Set Quantization (Required)
<input type="checkbox"/> Enter Rendezvous Point (Optional)	<input type="checkbox"/> Enter Exclusion Areas (Optional)
<hr/>	
Set Quantization	
(Position Cursor: Enter Data)	
Approximate Interval Between Course Changes (Hours) :	
Maximum Deviation From Mean Track	
To Starboard (N. Miles)	
To Port (N. Miles)	
Precision Option (Choose One. Position Cursor, Press "ENTER")	
<input type="checkbox"/> Low Resolution Solution / Short Run-Time	
<input type="checkbox"/> Medium Resolution Solution / Medium Run-Time	
<input type="checkbox"/> High Resolution Solution / Long Run-Time	

Figure 3-9: Data Form for SET QUANTIZATION

The interval between course changes determines the stages in the dynamic programming (DP) algorithm: decisions as to course and speed are made at the start of the transit and at integer multiples of the time interval thereafter until the transit finish time. This interval also represents the frequency with which the probabilities of detection per cell per stage are calculated. For example, a six hour interval means that the DP algorithm generates paths that maintain course and speed for six hours at a time and positions of enemy sensor platforms are calculated every six hours. The smaller the interval, the more accurate the modeling of the sensor scenario and the more flexibility the DP has to generate paths that avoid detection. A smaller interval also implies that the DP has to make more decisions in solving the surveillance avoidance problem so that runtime is larger. In determining what interval to input, the user can consider how often he is likely to want to make course changes as a function of the military situation and operational doctrine, the desired resolution for modeling the sensor scenario, and the amount of time available for allowing the dynamic programming algorithm to generate a solution.

Rather than quantize the entire map into a grid of square cells, a smaller rectangular area relevant to the transit is used as the problem space for the DP. The rectangle is oriented along the great circle path between start and end points, and the deviations to starboard and port of this mean path are used to determine the width of the rectangle. The user should choose the smallest deviations that he feels will yield enough room for maneuvering so as to reduce the computational requirements of the DP. For example, if a coastline lies within 500 nautical miles to starboard of the mean track, the user should input no more than 500 n. miles of starboard deviation so that the DP doesn't waste time considering and rejecting transitions that go through land. The user should also consider his time and fuel constraints in estimating how much deviation from mean track he should input.

The user also chooses a resolution option from the menu at the bottom of the SET QUANTIZATION form. This option determines the number of cells that are reachable from a generic cell if the ship travels at the

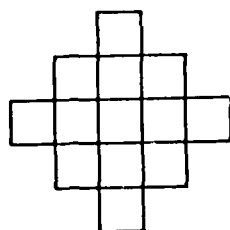
input maximum speed for one decision interval of time. The transition space for a generic cell is determined by drawing a circle that is centered around the center of the cell and has a radius equal to the maximum distance the ship can travel in the decision interval, then choosing the cell size so that the circle is approximated by one of the patterns in Fig. 3-10, depending on the option selected by the user. With finer resolution option there are more reachable cells, yielding more potential directions and speeds of transition. Finer resolution also yields more cells per unit area, meaning more cells in the problem grid, meaning longer runtime for the DP. The user should consider the trade-off between the higher accuracy that accompanies high resolution and the resulting increase in runtime.

The constraints for SET QUANTIZATION are that the three inputs must be nonnegative and the user must select a resolution option. SET QUANTIZATION is shown in Fig. 3-11.

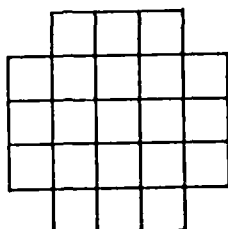
#### 3.1.3.3 RENDEZVOUS Module

The user has the option of entering one rendezvous point and date/time. The dynamic programming algorithm will then generate only paths that pass through the rendezvous point at the beginning of the stage containing the rendezvous time. Figures 3-12 and 3-13 show the summary and data form for RENDEZVOUS.

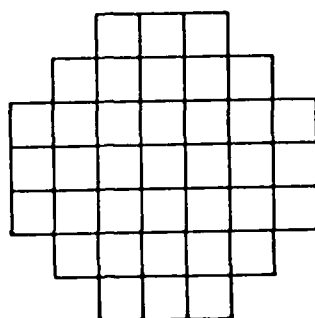
The constraints to be checked are that the latitude and longitude fall in appropriate ranges and directions, that the date is a valid date, and that the rendezvous point is on the map and in water. Note that checking the feasibility of passing through the rendezvous point and time with respect to the start and finish points of the transit is not possible at this point because the user can perform RENDEZVOUS before BASIC if he wishes.



COARSE RESOLUTION



MEDIUM RESOLUTION



FINE RESOLUTION

Figure 3-10: Resolution Options

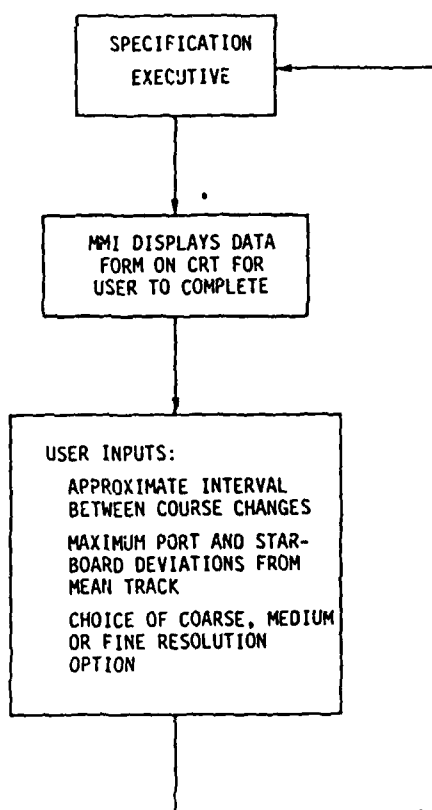


Figure 3-11: Summary of SET QUANTIZATION Module



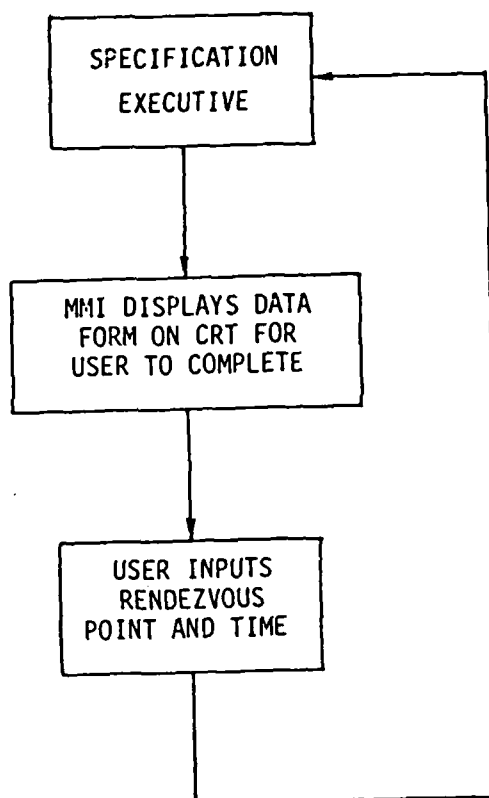


Figure 3-12: Summary of RENDEZVOUS Module

<input type="checkbox"/> General Instructions	<input type="checkbox"/> Exit ISAS
<input type="checkbox"/> Re-Specify Problem	<input type="checkbox"/> Proceed to Next Task
<hr/>	
SPECIFY A TRANSIT PROBLEM	
(Position Cursor: Press "ENTER")	
<input type="checkbox"/> Describe Basic Problem (Required)	<input type="checkbox"/> Set Quantization (Required)
<input checked="" type="checkbox"/> Enter Rendezvous Point (Optional)	<input type="checkbox"/> Enter Exclusion Areas (Optional)
<hr/>	
Enter Rendezvous Points	
(Position Cursor: Enter Data)	
Latitude	Longitude
	Date
	Time (GMT)
	/ /

Figure 3-13: Data Form for RENDEZVOUS

#### 3.1.3.4 EXCLUSION Module

The user can specify up to ten circular avoidance regions in the EXCLUSION module depicted in Fig. 3-14. Using the data form in Fig. 3-15, the user describes each circular region by inputting the center, radius, starting date/time, and ending date/time. If left blank, the start or end date/time defaults to the start or end date/time for the transit. Future extensions to EXCLUSION would be to allow the user to specify a friendly or unfriendly ship or a particular weather pattern to be avoided, and to allow noncircular regions to be specified by giving the user the means to draw regions on the graphics monitor.

Constraint checks are that latitudes, longitudes, dates, and times are valid inputs, all radii are positive, and end date/times are later than begin end/times.

#### 3.1.3.5 CHECK AND QUANTIZE Module

When the user signifies that he has finished the SPECIFICATION function and is ready to proceed to the next function, there are further constraint checks to be performed, and some quantization routines must be run, as shown in Fig. 3-16. The first checks are that the two required modules, BASIC and QUANTIZE, were properly completed. The same type of check is done for RENDEZ VOUS and EXCLUSION, if the user selected either of those options.

If the user entered a rendezvous point, some constraints checks that require information from the BASIC module are performed at this point. The rendezvous must occur during the transit, and it must be possible for the ship to pass through the rendezvous point at the right time without exceeding maximum speed or fuel constraints. These checks are performed using great circle distances.

The next ISAS task is to attempt to construct the problem grid, as outlined in Fig. 3-17. A grid coordinate system is constructed by treating the great circle through the transit start and end points as

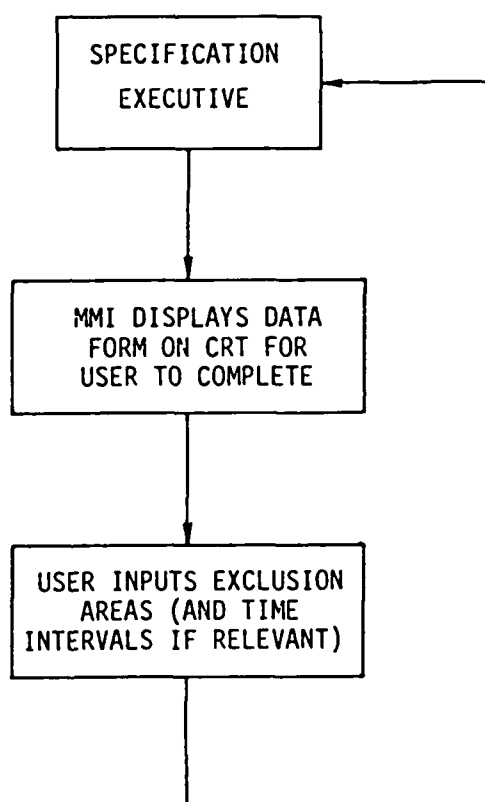


Figure 3-14: Summary of EXCLUSION Module

- General Instructions                      - Exit ISAS  
- Re-Specify Problem                      - Proceed to Next Task

---

SPECIFY A TRANSIT PROBLEM  
(Position Cursor: Press "ENTER")

- Describe Basic Problem (Required)      - Set Quantisation (Required)  
- Enter Rendezvous Point (Optional)      - Enter Exclusion Areas (Optional)

---

ENTER EXCLUSION AREAS

Enter up to ten circular exclusion areas and time intervals

	Center Latitude	Longitude	Size (N. Miles) Radius	From Date	Time(GMT)	Until Date	Time(GMT)
1				/	/		
2				/	/		
3				/	/		
4				/	/		
5				/	/		
6				/	/		
7				/	/		
8				/	/		
9				/	/		
10				/	/		

- Show Graphically

Figure 3-15: Data Form for EXCLUSION

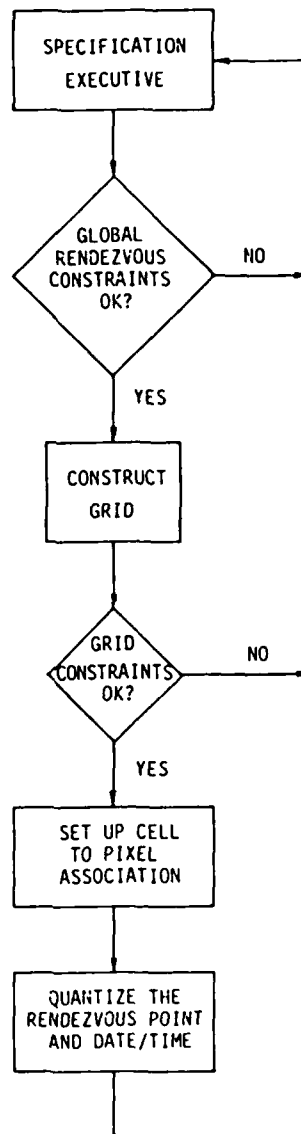


Figure 3-16: Summary of CHECK AND QUANTIZE Module

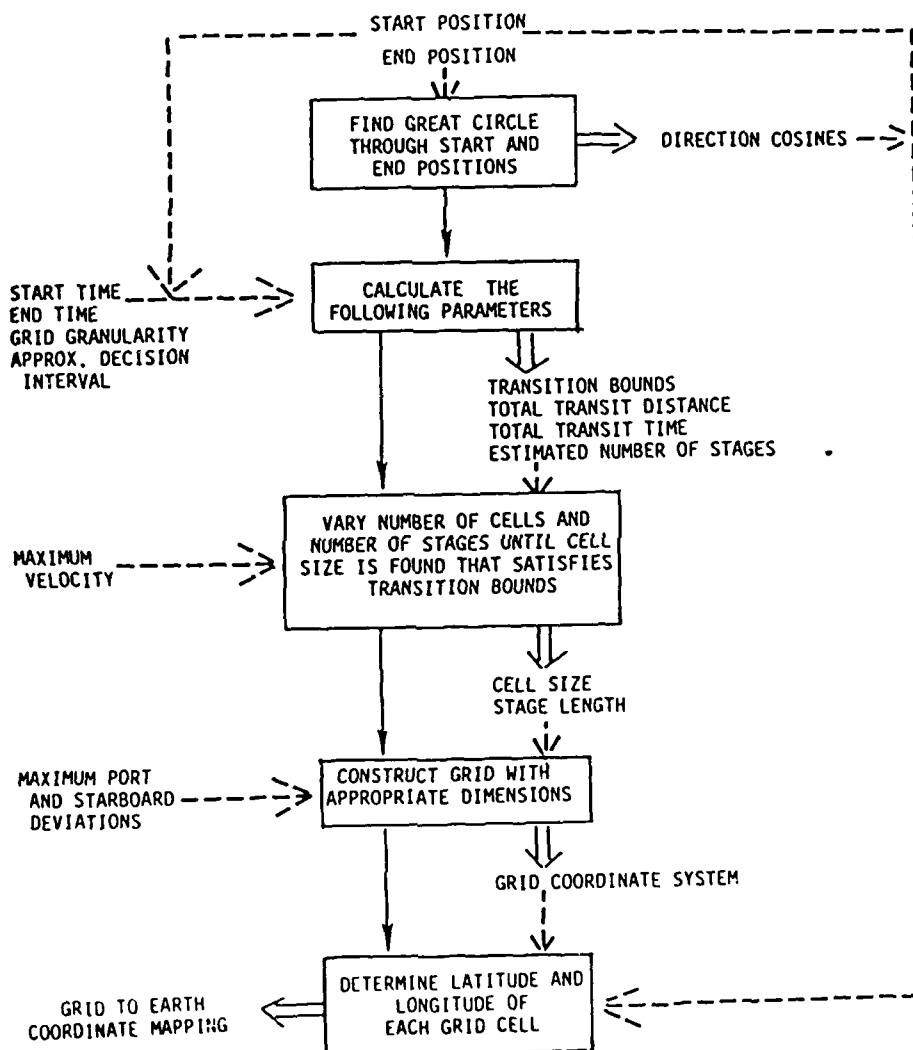


Figure 3-17: Construction of Problem Grid

the equator in the grid latitude/longitude system. Using the approximate interval between course changes as the starting guess, the number of stages and the number of cells from start to end point are varied until a combination is found that yields the transition space corresponding to the chosen resolution option. The difference between start and end longitude in the grid coordinate system is divided by the number of cells from start to end point to determine the size of each cell in units of degrees of longitude. The north and south boundaries of the grid in the new coordinate system are determined from the maximum deviations from mean track that were input by the user. The east and west of the grid are placed so that the start and end points are a few cells in from the end of the grid.

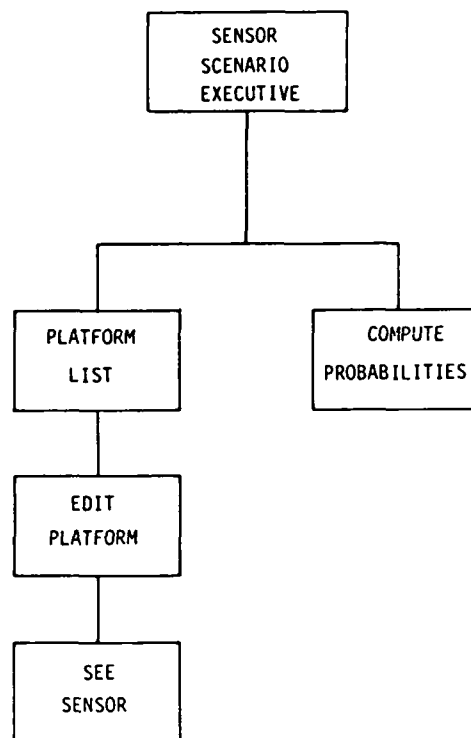
Due to array dimensions and runtime considerations, there are upper bounds on the number of allowable stages and cells. These constraints are checked in the course of constructing the grid. If a grid cannot be constructed within the constraints, the user must reduce the size of the problem by increasing the interval between course changes or choosing a coarser resolution option or breaking the transit into two or more legs and solving each leg separately.

The final operations performed in CHECK AND QUANTIZE are to construct a mapping between grid cells and graphics monitor pixels, and to quantize the rendezvous latitude/longitude and date/time into a grid cell and stage.

#### 3.1.4 SENSOR SCENARIO Function

This function provides the user interface needed to set up the sensor scenario specific to the current problem. Interfaces with existing data files and with generalized sensor models are provided. The capability to edit certain aspects of the sensor scenario are provided to the user to simulate updates to the data base that occur operationally as a result of newly received messages. A functional overview of the SENSOR SCENARIO function is given in Figure 3-18.





PLATFORM LIST	MENU OF SENSOR PLATFORMS IN SCENARIO
EDIT PLATFORM	EXAMINE/MODIFY DATA FOR SPECIFIC PLATFORM
SEE SENSOR	EXAMINE SPECIFIC SENSOR MODEL
COMPUTE PROBABILITIES	COMPUTE DETECTION PROBABILITIES PER CELL PER STAGE

Figure 3-18: Summary of SENSOR SCENARIO Function

#### 3.1.4.1 PLATFORM LIST Module

Upon entering SENSOR SCENARIO, the user is taken automatically into the module PLATFORM LIST. PLATFORM LIST reads in information concerning the sensor scenario from a default file, much like an operational ISAS system would receive sensor information by processing information contained in an intelligence data base. PLATFORM LIST provides the user with a list of platforms that are potential surveillance threats for the problem at hand, with name and type displayed for each platform. Also displayed are menu selections for entering each type of sensor platform. A sample form is shown in Fig. 3-19. By moving the cursor to the right position and pressing the appropriate function key, the user can delete a platform from the scenario, examine or edit information concerning a particular platform in the list, or add a new platform. Figure 3-20 gives a summary of PLATFORM LIST.

#### 3.1.4.2 EDIT PLATFORM Module

Depending on the type of platform selected - fixed, barrier search, transiting, or satellite - EDIT PLATFORM presents the user with one of the data forms depicted in Figs. 3-21 through 3-24, thus allowing the user to examine or edit the parameters of an existing platform in the scenario or to input all of the parameters describing a new platform. For the chosen sensor platform, the name, type, position, motion model (if it is a moving sensor), and list of on board sensors is displayed. The user can enter or change a parameter, add or delete a sensor from the platform, or elect to examine some particular sensor model. EDIT PLATFORM is summarized in Fig. 3-25.

As with the other data forms, the forms for EDIT PLATFORM have certain constraint checks associated with them. All latitudes, longitudes, dates, and times are checked for validity. Each platform must also have a name and at least one sensor on board. For the earth-based moving platforms, the ship type must match a known Soviet ship type in the data base, latitude/longitude pairs must be on the map and in the water, and transit speeds must be greater than zero but not greater than the physi-

- General Instructions
- Re-Specify Problem
- Exit ISAS
- Proceed to Next Task

---

EXAMINE / EDIT THE SENSOR SCENARIO

(Position Cursor: Press "ENTER" to examine/edit platform;  
Press "ADD" to create new platform; Press "REMOVE" to delete platform)

Platform Name	Platform Type
- New	Earth-Based, Fixed
- New	Earth-Based, Barrier Search
- New	Earth-Based, Transiting
- New	Satellite
<hr/>	
- Somalia SOSUS	Fixed, Earth Based
- Surveillance Ship 1	Barrier Search
- Surveillance Ship 2	Barrier Search
- Transiting Ship 1	Transitting
- Transiting Ship 2	Transitting
- Satellite 1	Satellite

Figure 3-19: Menu for PLATFORM LIST

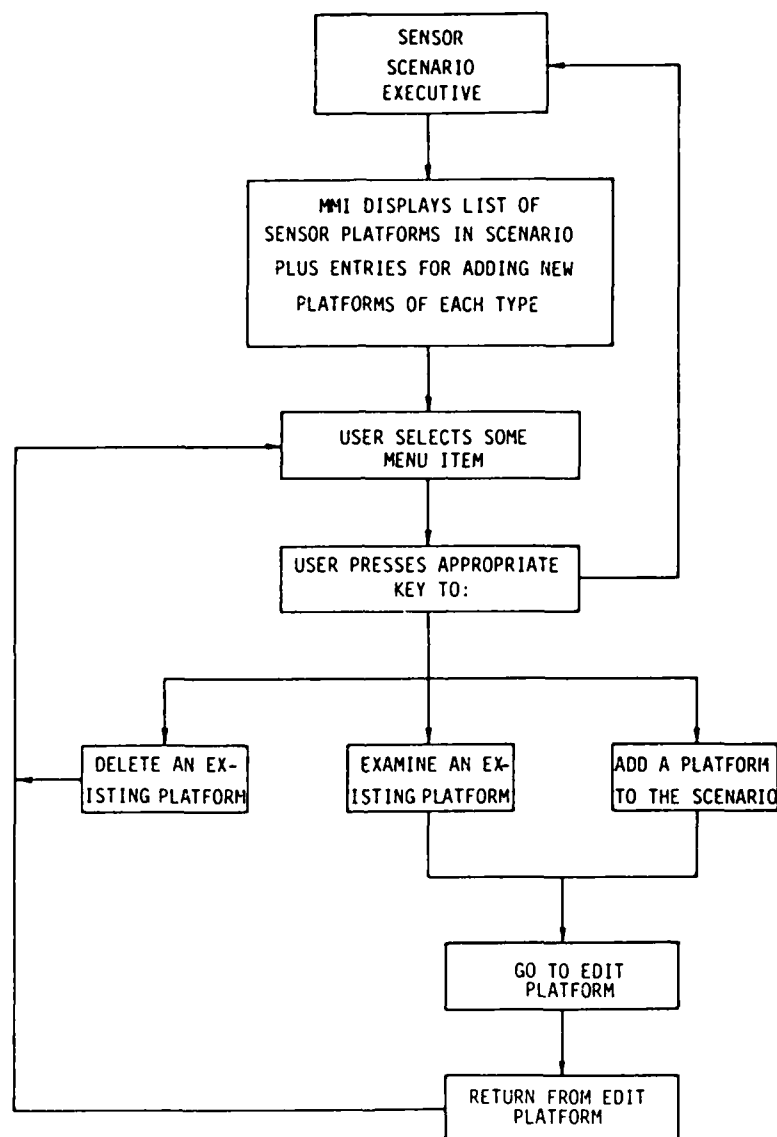


Figure 3-20: Summary of PLATFORM LIST Module

- General Instructions	- Exit ISAS
	- Form Completed

---

EARTH-BASED, FIXED PLATFORM  
(Position Cursor, Enter Data)

NAME

Location	Latitude	Longitude
----------	----------	-----------

---

(Position cursor: Press ENTER to examine sensor definition.  
Press REMOVE to delete sensor from platform.  
Press ADD to add sensor to platform.)

Sensors-

- SOSUS
- ESM
- OTH Radar
- Search Radar

Figure 3-21: Data Form for EARTH-BASED, FIXED PLATFORM

☐ General Instructions                      ☐ Exit ISAS  
☐ Form Completed

---

EARTH BASED, BARRIER SEARCH PLATFORM  
(Position Cursor: Enter Data)

Name

Type

	Latitude	Longitude	Speed	Date	Time
Start				/	/
Turn					

---

(Position cursor: Press ENTER to examine sensor definition.  
Press REMOVE to delete sensor from platform.  
Press ADD to add sensor to platform.)

Sensors-

<input type="checkbox"/> TASS	<input type="checkbox"/> ESM	<input type="checkbox"/> Search Radar
-------------------------------	------------------------------	---------------------------------------

Figure 3-22: Data Form for EARTH-BASED, BARRIER SEARCH PLATFORM

☐ General Instructions
 ☐ Exit ISAS  
☐ Form Completed

---

EARTH BASED, TRANSITING PLATFORM  
(Position Cursor, Enter Data)

Name .

Type .

	Position		Speed of	Date	Time
	Latitude	Longitude	Adv. (knots)		
1				/	/
2					
3					
4					
5					
6					

Note : To terminate transit - Enter no speed of advance for last position.  
 For repeating transit - Enter a final speed for return to first position

---

(Position cursor, Press ENTER to examine sensor definition;  
 Press REMOVE to delete sensor from platform;  
 Press ADD to add sensor to platform.)

Sensors-

☐ TASS
 ☐ ESM
 ☐ Search Radar

Figure 3-23: Data Form for EARTH-BASED, TRANSITING PLATFORM

General Instructions

Exit ISAS  
Form Completed

---

SATELLITE PLATFORM

(Position Cursor: Enter Data)

Name

Angle of Orbital Inclination (Degrees. S to N)

Orbital Period (Minutes)

Reference Equatorial Crossing Data -

Longitude

Date

Time

/ /

---

(Position cursor: Press ENTER to examine sensor definition;  
Press REMOVE to delete sensor from platform;  
Press ADD to add sensor to platform.)

Sensors-

NOR

IR

Figure 3-24: Data Form for SATELLITE PLATFORM



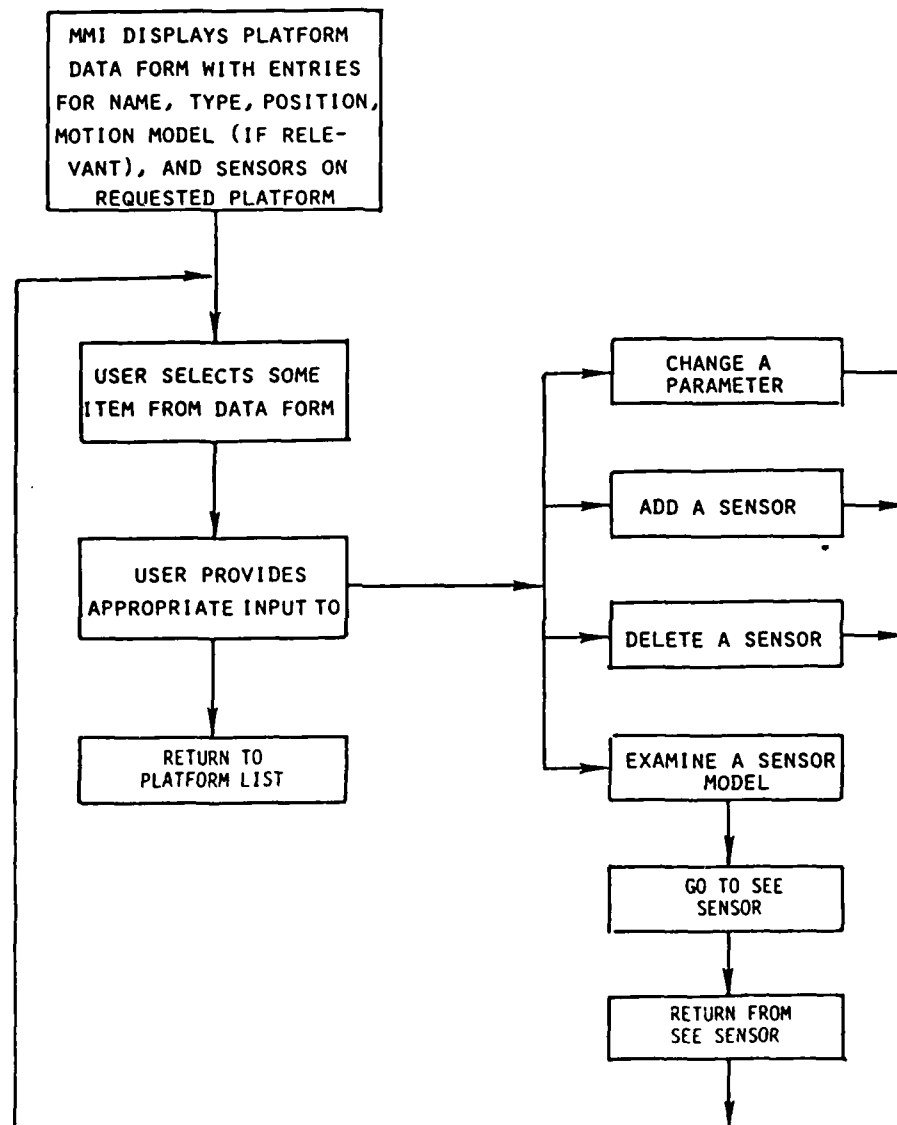


Figure 3-25: Summary of EDIT PLATFORM Module

cal maximum speed for the corresponding ship type. For the satellite platforms, inclination angle must be between 0 and 180 and period must be greater than 0.

The function SEE SENSOR (Fig. 3-26) displays the detection probability model for a requested generic sensor type. The user is not allowed to change any of the sensor performance parameters; he can only examine the model to understand the surveillance capabilities of any particular sensor. Detection models for earth-based sensors are in the form of a series of concentric circles, with the user being shown the radius and associated detection probability for each circular band. All of the sensor models are shown in Figs. 3-27 through 3-33.

The COMPUTE PROBABILITIES module calculates and stores the probability of nondetection associated with each cell of the problem grid for each stage of the quantized time. The probabilities are calculated based on the generic sensor models and the predicted position of each sensor platform as a function of time. Probability of nondetection per cell and stage is calculated and stored as a real number. (ESM detection probabilities are stored separately because our model is that line-of-sight emissions are not used so that avoidance of ESM detection can be achieved through EMCON rather than by maneuvering the ship.) Independence of sensors is assumed in calculating the probability of detection value for a cell observed by more than one sensor at a given time; i.e., if a cell is in the coverage areas of  $n$  distinct sensors, then it is assumed that the sensors act independently of one another rather than to alter the overall probability of detection in the cell by cooperating. The negative logarithm of the probability of nondetection for each cell at each stage is calculated, multiplied by a scaling factor, rounded off to an integer, and stored. This conversion to integers allows the dynamic programming algorithm to use integer addition rather than floating point multiplication in its calculations. Negative numbers are then inserted as required to distinguish both land mass areas and exclusion areas that a path cannot pass through. This module is described in Fig. 3-34.

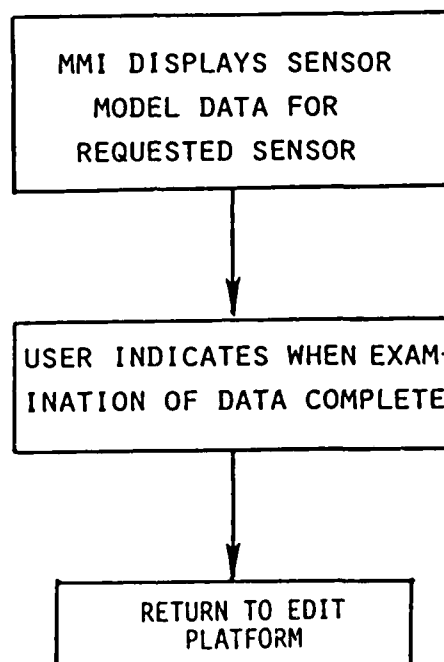


Figure 3-26: Summary of SEE SENSOR Module

- General Instructions

- Exit ISAS  
- Examination Complete

---

EARTH BASED SENSOR

Sensor Type SOSUS

	Radius (In N. Miles)	Probability of Detection/Hour
1	1000.00	0.01000
2	125.000	0.20000
3	115.000	0.10000
4	95.0000	0.30000
5	85.0000	0.15000
6	65.0000	0.50000
7	55.0000	0.20000
8	35.0000	0.70000
9	25.0000	0.30000
10	10.0000	0.90000

Figure 3-27: Sensor Model for SOSUS

```

- General Instructions
- Exit ISAS
- Examination Complete

```

## EARTH BASED SENSOR

**Sensor Type : ESM**

	Radius (In N. Miles)	Probability of Detection/Hour
1	70.0000	0.10000
2	50.0000	0.40000
3	35.0000	0.90000
4		
5		
6		
7		
8		
9		
10		

**Figure 3-28: Sensor Model for ESM**

- General Instructions

- Exit ISAS  
- Examination Complete

---

EARTH BASED SENSOR

---

Sensor Type : O.T.H. Radar

	Radius (In N. Miles)	Probability of Detection/Hour
1	2000.00	0.30000
2	2500.00	0.50000
3	1200.00	0.30000
4	900.000	0.00000
5		
6		
7		
8		
9		
10		

Figure 3-29: Sensor Model for OTH

\_ General Instructions

\_ Exit ISAS

\_ Examination Complete

---

EARTH BASED SENSOR

---

Sensor Type . Search Radar

	Radius (In N. Miles)	Probability of Detection/Hour
1	52.0000	0.10000
2	37.0000	0.30000
3	26.0000	0.80000
4		
5		
6		
7		
8		
9		
10		

Figure 3-30: Sensor Model for Search Radar

```

- General Instructions
- Exit ISAS
- Examination Complete

```

**EARTH BASED SENSOR**

**Sensor Type : TASS**

	Radius (In N. Miles)	Probability of Detection/Hour
1	94.0000	0.20000
2	86.0000	0.00000
3	63.0000	0.50000
4	57.0000	0.05000
5	32.0000	0.70000
6	28.0000	0.10000
7	10.0000	0.90000
8		
9		
10		

**Figure 3-31: Sensor Model for TASS**



General Instructions                      Exit ISAS  
   Examination Complete

---

SATELLITE BASED SENSOR

Sensor Type :	R.O.R.	
Satellite Scan Direction		Left
Distance From Orbital Plane to Coverage (N. Miles) :	300.000	
Distance Across Coverage (N. Miles)	300.0	
Detection Probability/Hour	0.000	

Figure 3-32: Sensor Model for Satellite ROR



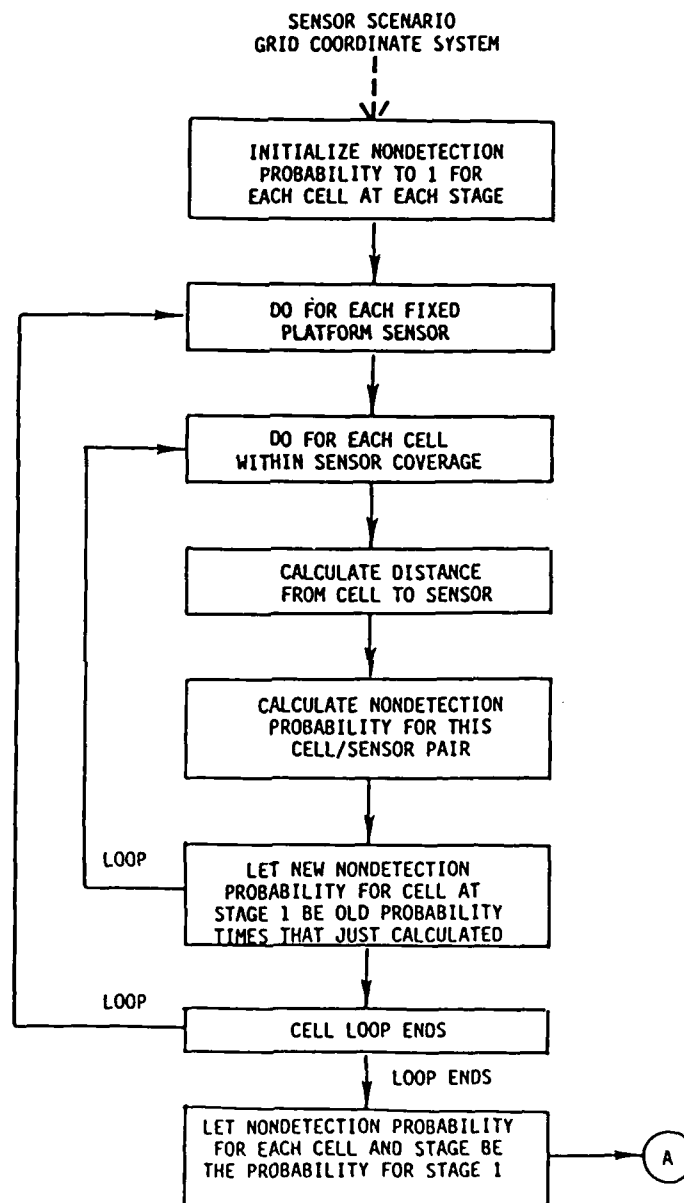


Figure 3-34: Summary of COMPUTE PROBABILITIES Module

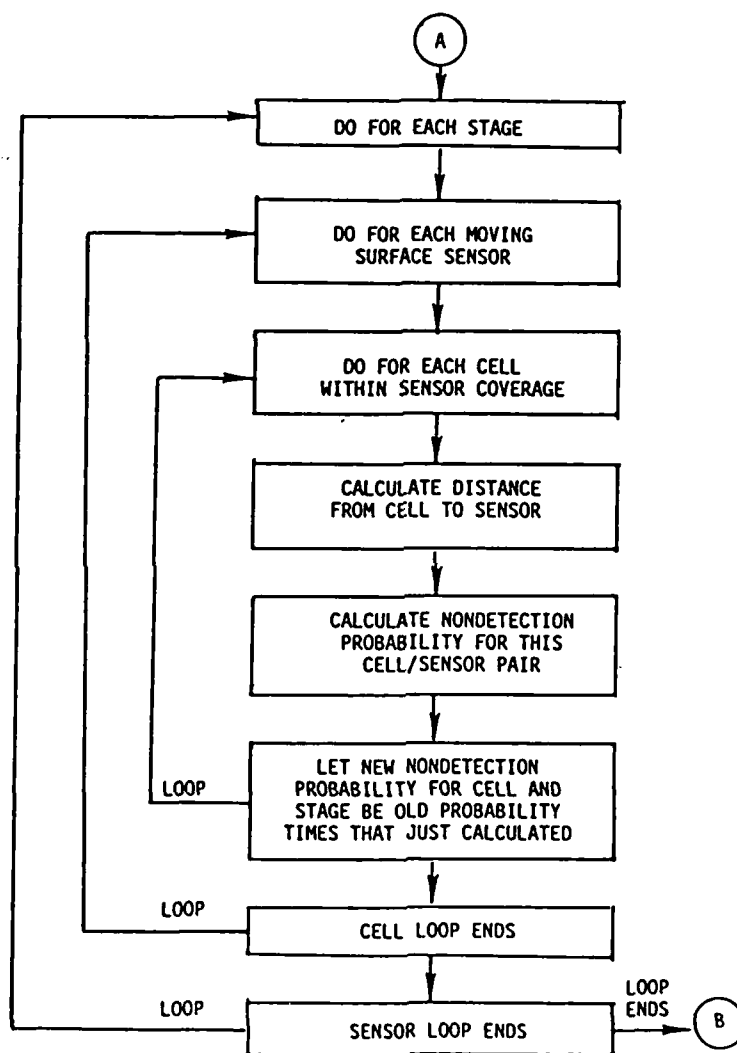


Figure 3-34: Summary of COMPUTE PROBABILITIES Module (Cont.)

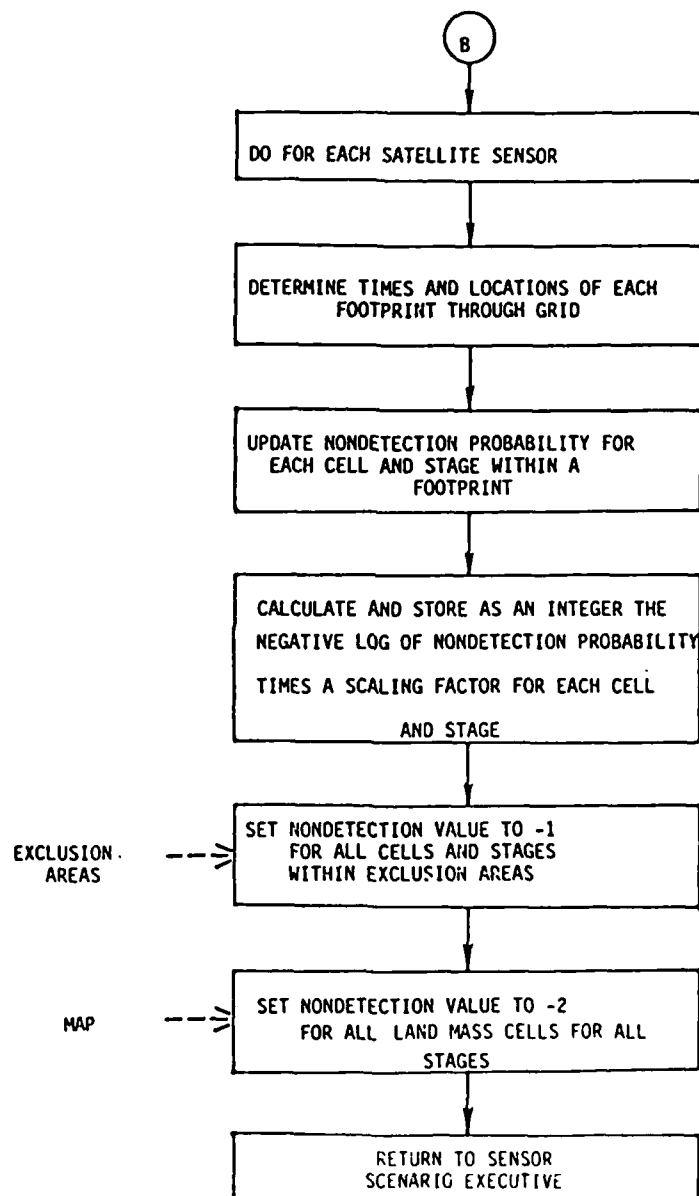


Figure 3-34: Summary of COMPUTE PROBABILITIES Module (Cont.)

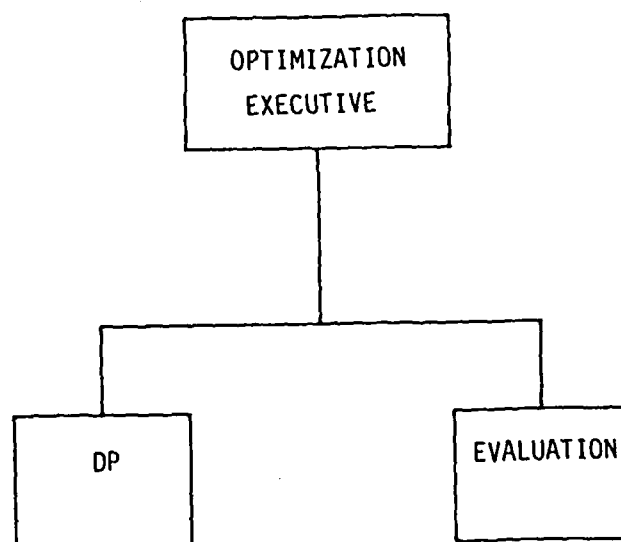
### 3.1.5 OPTIMIZATION Function

The OPTIMIZATION function generates and evaluates an optimal path for the problem specified by the user. Optimization is achieved through use of the dynamic programming technique. The solution process is divided into two modules, DP and EVALUATION, as seen in Fig. 3-35. The DP function is called by OPTIMIZATION to generate a set of transition decisions that define a solution path. EVALUATION then traces through and analyzes the path.

#### 3.1.5.1 DP Module

The DP function executes a forward dynamic programming algorithm. General descriptions of the dynamic programming technique are given in Section 1.3 and in Appendix A. The following is a brief description of the specific formulation of DP that is used in ISAS for the surveillance avoidance problem and that is described in Fig. 3-36.

The stage variable is time: a fixed decision interval is established such that a new decision is required at the end of each decision interval. The state variable is position: for any given stage the DP algorithm computes the optimal cumulative cost to get from the cell that the evading ship started in to each of the cells that the evader could be in at that stage. The decision variable is choice of feasible transition into a cell (transition choice implicitly defines course and speed for the transition): a transition can pass through several cells, depending on cell size, decision interval, and speed of the evading ship. Transition cost is cumulative probability of detection of the transition plus a Lagrange multiplier times the amount of fuel consumed by the transition. In general, a few iterations over the value of the Lagrange multiplier will be required before the final solution is found.



DP	DYNAMIC PROGRAMMING PROCEDURE FOR PATH OPTIMIZATION
EVALUATION	EVALUATE A PATH

Figure 3-35: Summary of OPTIMIZATION Function

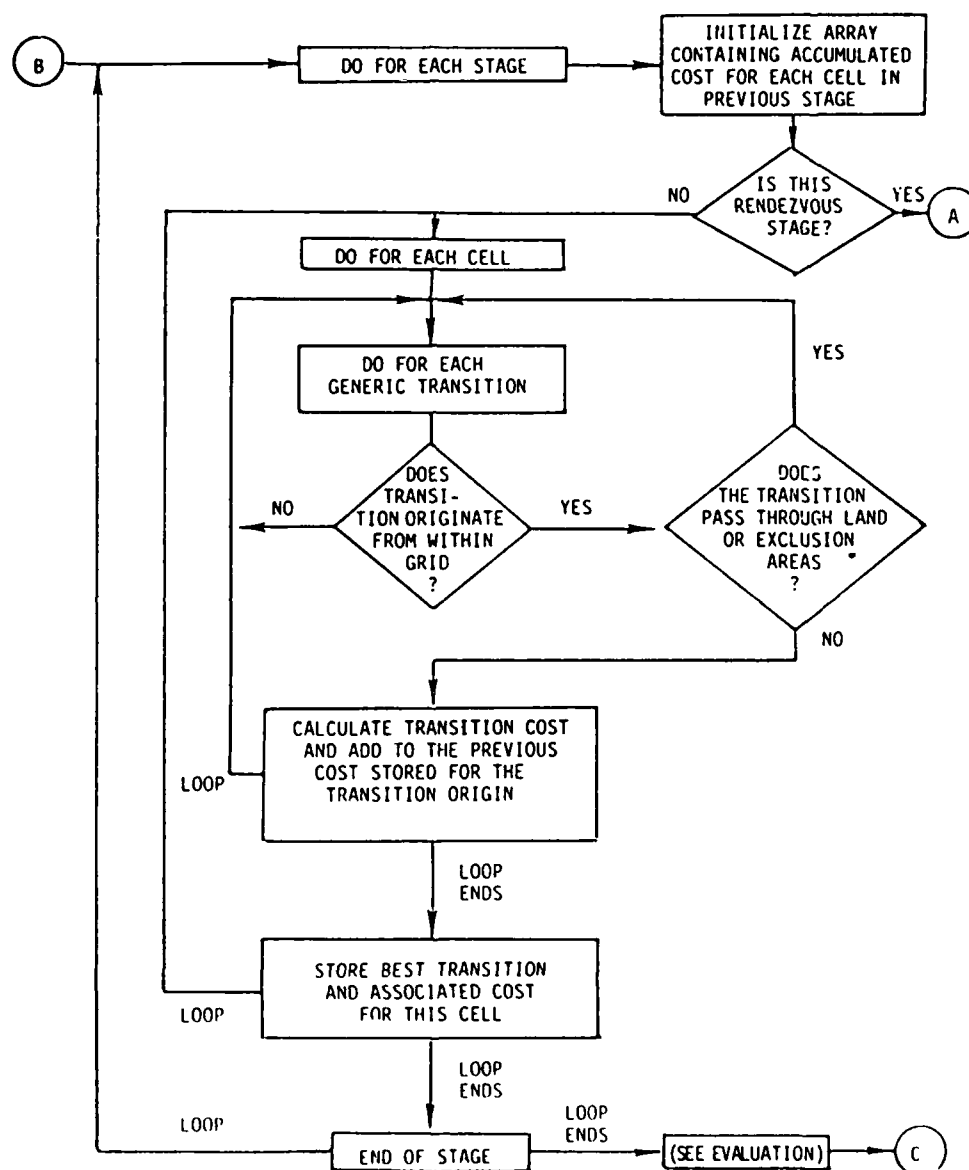


Figure 3-36: Summary of DP Module



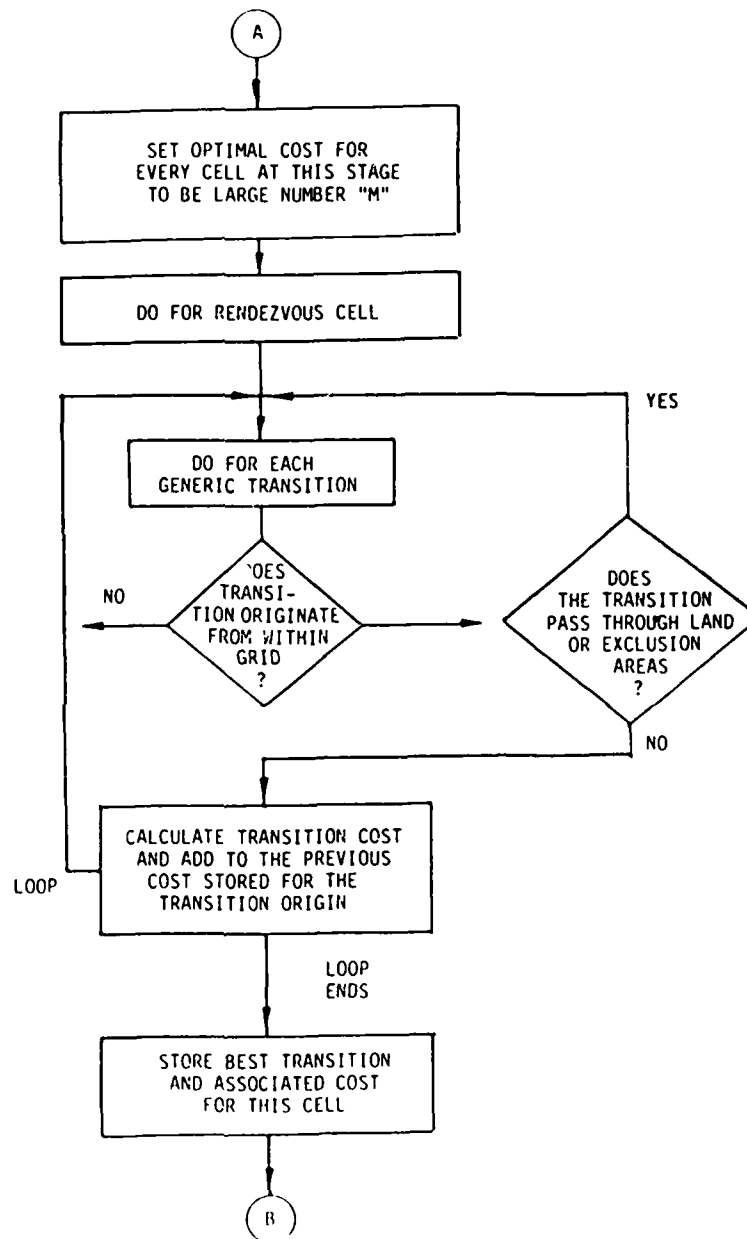


Figure 3-36: Summary of DP Module (Cont.)

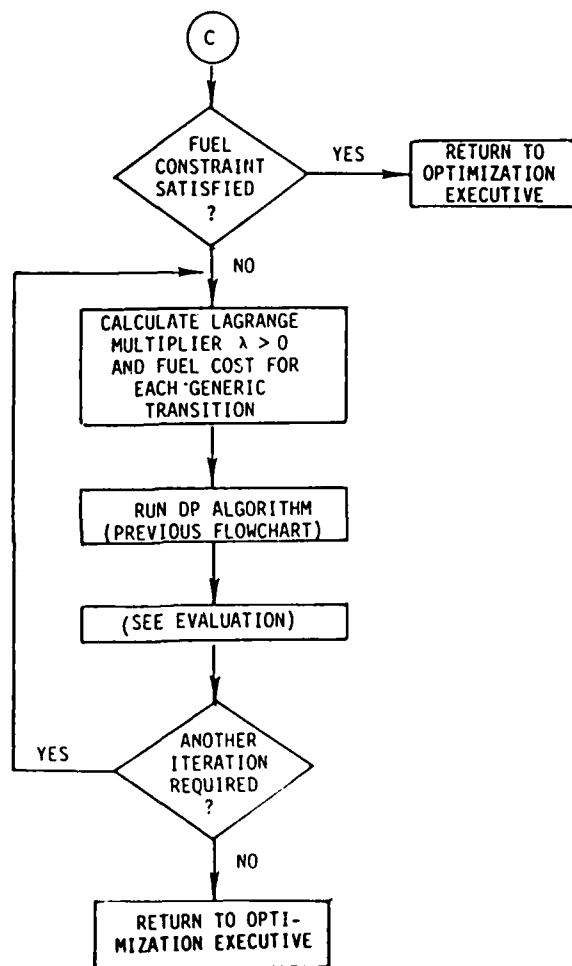


Figure 3-36: Summary of DP Module (Cont.)

#### 3.1.5.2 EVALUATION Module

The DP module finds the optimal transition into every cell at every stage. Using the information generated by the DP module, EVALUATION works back through the optimal transitions, starting with the optimal transition into the destination cell at the final stage, to construct the entire optimal path. For each transition along the optimal path, EVALUATION calculates the real-valued probability of detection, course, speed, and fuel use, and also determines whether any ESM sensor is close enough to detect line-of-sight emissions. This data is stored for future display, and is used to calculate cumulative probability of detection and fuel use for each stage of the transit. Figure 3-37 shows the EVALUATION module.

#### 3.1.5.3 Executive Operations

Before initiating the DP module, the OPTIMIZATION executive has the MMI display the form in Fig. 3-38 to the user, so that the user knows what will happen next. The first iteration of the DP module has no penalty on fuel use and finds the global, unconstrained optimal path. The path is evaluated and drawn on the graphics monitor, and the user is shown the form in Fig. 3-39. This information helps the user to decide whether to examine this path more closely later in ISAS. If the unconstrained optimum satisfies the fuel constraint, then it is the desired solution and no further iterations are necessary. If the fuel constraint is violated, then an initial guess of the Lagrange multiplier is calculated, fuel penalty is introduced into the cost function, and DP is run again in hopes of finding a fuel feasible path. With the generation of each path, the user is shown the picture of the path and the brief summary information concerning the path. The Lagrange multiplier is iteratively increased until a feasible path is found. Having found upper and lower bounds on the Lagrange multiplier, a bisection method is used to look for the value of the Lagrange multiplier that yields the best solution that satisfies the fuel constraint. The iteration ends when this value is found or when a given upper bound on the number of iterations is reached. If a solution has been found, the user is

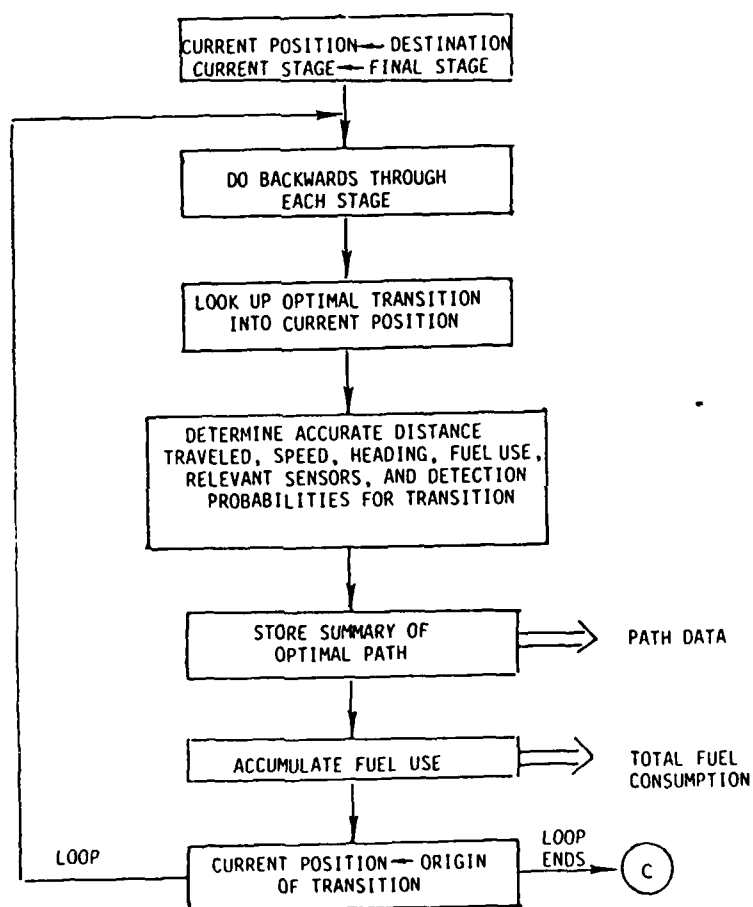


Figure 3-37: Summary of EVALUATION Module

RUN THE PATH OPTIMIZATION ALGORITHM  
(Wait for prompt; keyboard will not respond)

The path optimisation algorithm is currently running. The algorithm may generate several paths while searching for the optimum. ISAS will notify you when each path is generated and will provide you the name that it has assigned to that path. Please wait.

Figure 3-38: Initial OPTIMIZATION Form

Run the Path Optimization Algorithm  
(Wait for prompt; keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name : path1  
Cumulative Probability of Detection : 0.000  
Cumulative Fuel Use (% Capacity) : 39.78

Figure 3-39: Name Path Form

presented with the form shown in Fig. 3-40, otherwise the form in Fig. 3-41 is displayed.

#### 3.1.6 CREATE PATH Function

The CREATE PATH function is an unimplemented function that would enable a user to either enter a complete path or to modify an existing path. The function would contain a module for interaction with the user, preferably by allowing the user to mark way points on the graphics monitor and type in speeds of advance, so that the user can enter his path. The path would then be analyzed by an evaluation module to compute probability of detection per leg and cumulatively, cumulative fuel use, course and speed per leg, and course change dates/times, as is done for the paths generated automatically by the dynamic programming algorithm. This capability could be used to augment the path optimization by allowing the user to modify a path that was automatically generated to smooth it out, stay further away from land masses or enemy ships, to evaluate the result of performing a training maneuver along the transit, or any other reason for testing a modification to the path. The user could also bypass the automatic path generation entirely and use CREATE PATH to choose his own surveillance avoidance path and, using the evaluation module for feedback, alter the path until acceptable surveillance avoidance performance is achieved. The current ISAS effort did not require the capability for the user to specify paths. It is felt that such a function is desirable and useful, but time and budget constraints did not allow for implementation of this capability.

#### 3.1.7 SHOW RESULTS Function

The SHOW RESULTS function provides the user access to the evaluation results for any available route that has been created by the OPTIMIZATION function. Results concerning sensor coverage, type and degree of detection probability along a route, fuel usage, and course instructions will be available for examination. A functional overview of SHOW RESULTS is given in Fig. 3-42. Upon entering SHOW RESULTS, the user is presented with the a menu of the three capabilities of the function, as

<input type="checkbox"/> General Instructions	<input type="checkbox"/> Exit ISAS
<input type="checkbox"/> Re-Specify Problem	<input type="checkbox"/> Proceed to Next Task

---

Run the Path Optimisation Algorithm  
(Position Cursor; Press "ENTER")

The path optimisation algorithm has been completed. path6  
has been determined to be the optimal path and has been  
renamed "optimum". Please choose an item from the menu above.

Figure 3-40: OPTIMIZATION Ended Successfully Form



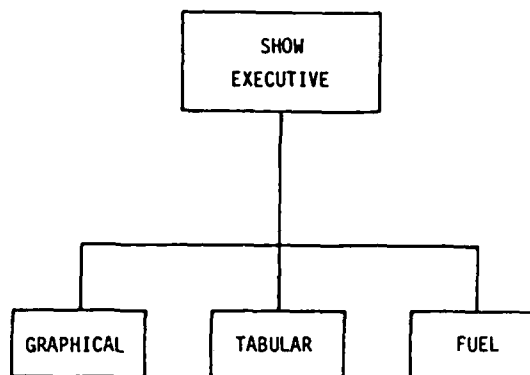
☐ General Instructions                      ☐ Exit ISAS  
☐ Re-Specify Problem                      ☐ Proceed to Next Task

---

Run the Path Optimisation Algorithm  
(Position Cursor: Press "ENTER")

The path optimisation algorithm has been completed.  
None of the paths generated satisfy the fuel constraint.  
However, these paths are available for inspection.  
Please choose an item from the menu above.

Figure 3-41: OPTIMIZATION Failed Form



GRAPHICAL	DISPLAY ROUTE INFORMATION (GRAPHICAL)
TABULAR	SUMMARIZE HEADING AND SPEED INSTRUCTIONS FOR PATH
	SUMMARIZE DETECTION PROBABILITIES ALONG PATH
FUEL	DISPLAY DETECTION PROBABILITY VERSUS FUEL USE

Figure 3-42: Summary of SHOW RESULTS Function

- General Instructions
- Re-Specify Problem
- Exit ISAS
- Proceed to Next Task

---

SHOW PATH EVALUATION RESULTS  
(Position Cursor: Press "ENTER")

- Transit Display (Graphical)
- Transit Summary (Tabular - Hard Copy)
- Graph of Fuel Consumption vs. Detection

Figure 3-43: Menu for SHOW RESULTS

seen in Fig. 3-43. The user selects one of the three modules, each of which is now described.

#### 3.1.7.1 GRAPHICAL Module

If the user requests a graphical display of a path, he is presented with a menu of the paths available for inspection, as shown in Fig. 3-44. The user selects a path for graphical display. The initial display on the graphics monitor depicts the sensor scenario for the first stage of the transit (the first stage corresponding to the first transition or leg of the path). Figure 3-45 has an example of such a display. The entire transit path is drawn, with the small squares at each end denoting the start and end points. The triangle colocated with the leftmost square indicates the current position of the evading ship. The double line extending from the current ship position indicates the distance to be traveled on the first leg. Probability of detection over the entire problem grid is shown by shading the detection regions with different colors. The blue regions are water with no probability of detection, bright red is high detection risk, and the shades of purple in between indicate intermediate detection probabilities.

In conjunction with the graphical display, the user is presented with the form in Fig. 3-46 that contains textual information about the image on the graphics monitor. The information tells the user what stage, date, and time the image corresponds to, and numerical status data for this point along the transit. The form also provides the user the ability to request to see the stage just before or after the one being viewed presently, or to jump to any stage or date/time of interest. The summary of this module is given in Fig. 3-47.

#### 3.1.7.2 TABULAR Module

Also available is the option to get a hard copy summary of any path. Upon choosing the TABULAR option, the user is presented a menu of available paths, as in the GRAPHICAL module. The user selects a path, and a table containing the textual information from GRAPHICAL for each

- General Instructions
- Re-Specify Problem
- Exit ISAS
- Proceed to Next Task

---

SHOW PATH EVALUATION RESULTS  
(Position Cursor: Press "ENTER")

- X Transit Display (Graphical)
- Transit Summary (Tabular - Hard Copy)
- Graph of Fuel Consumption vs. Detection

---

TRANSIT DISPLAY (Graphical)  
(Position Cursor: Press "ENTER")

Path Name

\_path1  
\_path2  
\_path3  
\_path4  
\_path5  
\_OPTIMUM

Figure 3-44: GRAPHICAL Path Menu

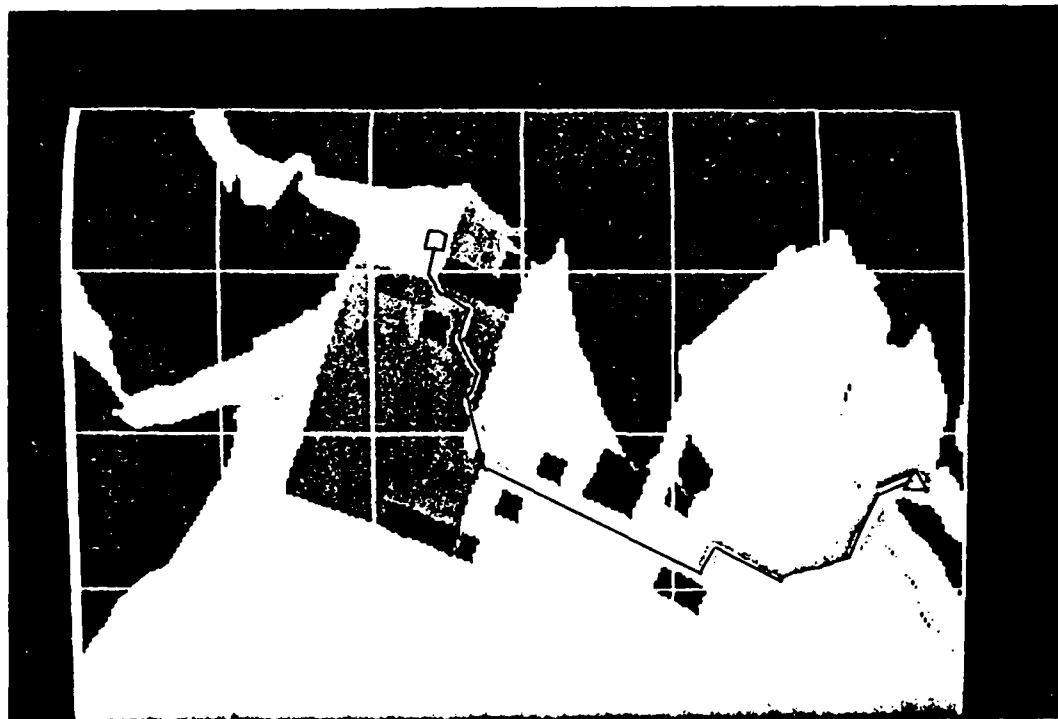


Figure 3-45: Sample Graphical Display



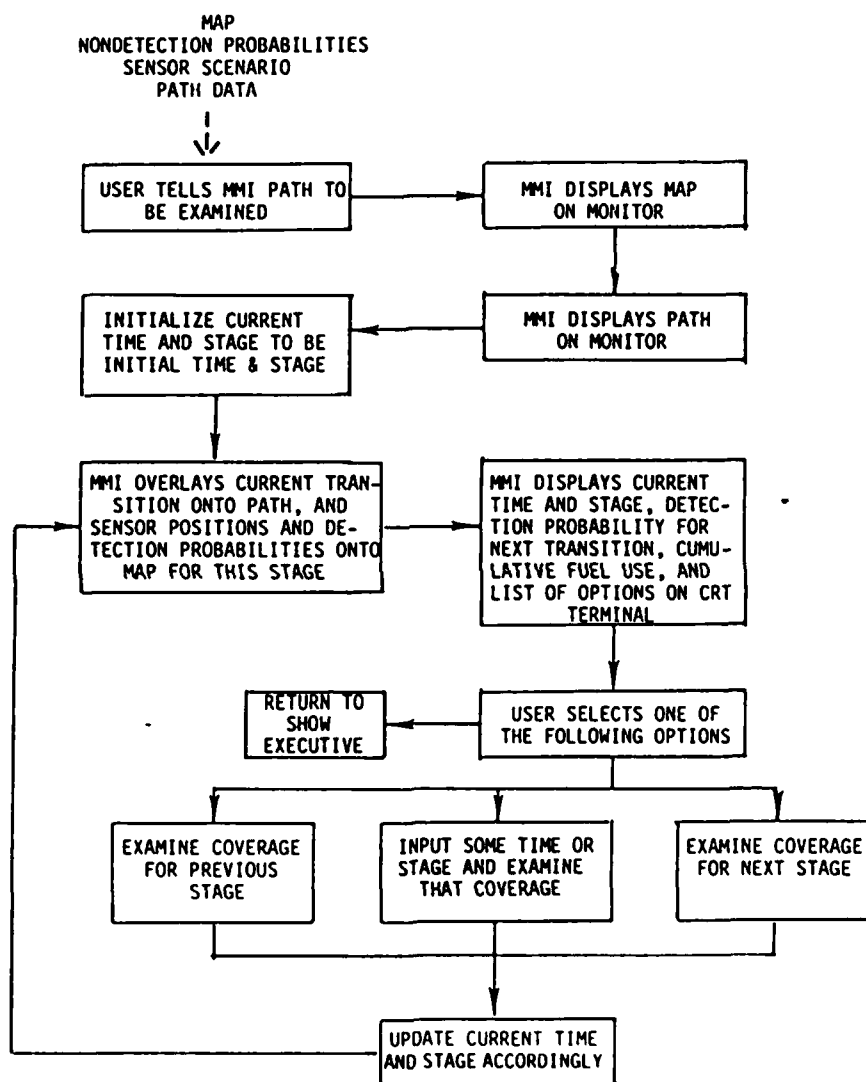


Figure 3-47: Summary of GRAPHICAL Module



stage of the transit is written to a file. The user can thus obtain a printout that summarizes all of the course and speed instructions and other data corresponding to any available path of interest. Figure 3-48 has an example of the output provided by TABULAR.

### 3.1.7.3 FUEL Module

FUEL, an as yet unimplemented module, would produce a graph of cumulative probability of detection versus total fuel consumption for the problem of interest. The plot would use data points generated during the Lagrange multiplier iterations of OPTIMIZATION. If there were not enough data points available to plot this function when requested, the user would be warned that further calculations were necessary and would be given a chance to exit FUEL if so desired. If the user approved the additional calculations, FUEL would run the dynamic programming algorithm for several additional Lagrange multipliers to generate the necessary data points. When enough data points existed, the plot would be drawn. FUEL would allow the user to easily see the possible tradeoffs between fuel use and detection probability. A path that has detection probability slightly higher than the optimum may use significantly less fuel, making it the preferred path to users who are concerned with fuel consumption.

## 3.2 MAN-MACHINE INTERFACE SOFTWARE

All interaction between the user and ISAS is handled by the man-machine interface. This module keeps the user informed of his progress through the program, and allows him to enter data and select options. This interaction is structured through the use of menus and forms. This section will discuss the interface, both as it appears to the user, and as it is implemented in ISAS.

Surveillance Avoidance  
System Software

Final Report  
Section 3

ISAS: Path Tabular Summary										
STAGE #	DATE	TIME (GMT)	POSITION		HEADING (DEG)	SPEED OF ADVANCE (KNOTS)	CUM. FUEL (% CAP.)	DETECTION PROBABILITY		LOS EMIS. SAFE?
			LAT	LONG				CUM.	NEXT	
1	7/26/83	1200	7.0n	97.0e	253.88	20.59	1.46	.000	.000	YES
2	7/26/83	1909	6.3n	94.6e	208.69	14.56	1.97	.000	.000	YES
3	7/27/83	0217	4.8n	93.8e	208.62	14.56	2.48	.000	.000	YES
4	7/27/83	0926	3.3n	93.0e	208.57	14.56	3.00	.000	.000	YES
5	7/27/83	1635	1.7n	92.1e	253.14	20.50	4.45	.000	.000	YES
6	7/27/83	2344	1.0n	89.8e	252.70	20.45	5.91	.000	.000	YES
7	7/28/83	0652	.3n	87.5e	297.78	14.32	6.42	.000	.000	YES
8	7/28/83	1401	1.1n	85.9e	297.44	14.32	6.94	.000	.000	YES
9	7/28/83	2110	1.9n	84.4e	297.07	14.32	7.45	.000	.000	YES
10	7/29/83	0418	2.7n	82.9e	206.87	14.56	7.97	.000	.000	YES
11	7/29/83	1127	1.1n	82.1e	296.66	14.23	8.48	.182	.182	YES
12	7/29/83	1836	1.9n	80.6e	296.24	14.23	9.00	.182	.000	YES
13	7/30/83	0145	2.6n	79.1e	295.81	14.23	9.51	.182	.000	YES
14	7/30/83	0853	3.4n	77.5e	295.35	14.23	10.03	.182	.000	YES
15	7/30/83	1602	4.1n	76.0e	294.87	14.23	10.54	.182	.000	YES
16	7/30/83	2311	4.8n	74.5e	294.38	14.23	11.05	.182	.000	YES
17	7/31/83	0619	5.5n	72.9e	293.86	14.23	11.57	.453	.331	YES
18	7/31/83	1328	6.2n	71.3e	293.33	14.23	12.08	.453	.000	YES
19	7/31/83	2037	6.9n	69.8e	292.77	14.23	12.60	.453	.000	YES
20	8/ 1/83	0345	7.5n	68.2e	337.81	20.39	14.05	.453	.000	YES
21	8/ 1/83	1054	9.8n	67.3e	337.17	20.45	15.51	.472	.035	YES
22	8/ 1/83	1803	12.0n	66.3e	21.61	14.56	16.02	.491	.035	YES
23	8/ 2/83	0112	13.6n	67.0e	336.63	20.53	17.48	.491	.000	YES
24	8/ 2/83	0820	15.9n	66.0e	21.34	14.56	17.99	.491	.000	YES
25	8/ 2/83	1529	17.5n	66.6e	270.00	.20	18.01	.491	.000	YES
26	8/ 2/83	2238	17.5n	66.6e	270.00	.20	18.04	.491	.000	YES
27	8/ 3/83	0546	17.5n	66.6e	270.00	.20	18.06	.491	.000	YES
28	8/ 3/83	1255	17.5n	66.6e	291.12	14.53	18.57	.491	.000	YES
29	8/ 3/83	2004	18.1n	64.9e	335.62	20.58	20.03	.491	.000	YES
30	8/ 4/83	0313	20.4n	63.8e	20.49	14.56	20.54	.491	.000	YES
31	8/ 4/83	1021	22.0n	64.5e	.00	.00	20.56	.660	.331	YES
Final	8/ 4/83	1730	22.0n	64.5e			20.56	.660		

Figure 3-48: Example of Tabular Summary

### 3.2.1 User Interaction with the Interface

As stated above, all interaction with ISAS is mediated by the interface, which displays information and accepts commands through a set of forms and menus. At each stage in the program there is either a data form or menu displayed on the user's terminal screen. A menu may display information and offer a choice of options which the user can select. A data form can do all the things a menu can do, plus it allows the user to input data.

When the user is faced with a menu, there is really relatively little he can do. There are several keys which he can press to move a cursor around the screen; these are labeled with arrows indicating the direction of motion. The cursor will only move to certain positions, those adjacent to pieces of text which describe options he may choose. At any such location, he can press one of four keys to register a choice. Those four keys are labeled ENTER, HELP, ADD, and REMOVE. ENTER is used for most menu selections; HELP provides an instructive message for the current task within ISAS, such as the one for QUANTIZATION that is shown in Fig. 3-49; ADD and REMOVE are only occasionally relevant, such as in EDIT PLATFORM for adding or removing sensors from a platform.

A convenient feature of the menus in the interface is that multiple layers of menus are displayed on the same screen whenever appropriate and feasible. For example, three levels of menus were seen in Fig. 3-44. The top-level menu is the usually present menu that allows the user to get general help, exit, respecify, or exit. The second level has the three modules within SEE RESULTS, and the third level is a menu of available paths. The user is free to move up and down through these levels in making his next function choice. This approach is easier to work with than if a single menu existed on each screen so that the user had to move through several screens to get up to the menu that he wanted.

\_ Return To Previous Location In ISAS

\_ Exit ISAS

---

HELP

(Position Cursor; Press "ENTER")

TOPIC: SET QUANTIZATION

The path optimization algorithm used by ISAS requires that space and time be quantized into a finite number of cells and time intervals. The SET QUANTIZATION task elicits information from you as to the required resolution of the quantization. The approximate interval between course changes is the smallest unit of time used by the algorithm. For a larger interval, the optimization algorithm will have fewer course change decisions to make and will therefore run faster, at the cost of a solution that is less accurate because of the coarseness of the model of time. Maximum deviation from mean track in either direction should be made as small as is acceptable. This limits the search space of the algorithm, thus reducing run-time. The precision option affects the size of the cells used by the algorithm. Low resolution yields a fewer number of larger cells, meaning that space is modeled coarsely, and results in faster run-time of the algorithm. High resolution yields a larger number of smaller cells. The model obtained with this option will be more accurate, possibly yielding a better surveillance avoidance path at the cost of increased run-time. Medium resolution is between these extremes in both accuracy and run-time.

Figure 3-49: Help Message for QUANTIZATION

A data form allows slightly more flexibility. The distinguishing feature of a data form is that it has input fields. These appear on the data form as boxes in reverse video. Using the arrow keys, the user can also move the cursor to these input fields. Once there, he may type appropriate data into the box. There are two more special keys to aid the user in editing inputs. The DELETE CHARACTER key removes the character directly under the cursor. The DELETE FIELD key removes all the characters, starting with the one under the cursor, up to the end of the field. There is also the normal DELETE key on the terminal, which removes the character before the cursor.

The user may move around the data form and enter data in any order. ISAS only gets to see the data when the user signals that the entire form is complete. As much as possible, related inputs are grouped on the same form. This allows the user to see his responses to several questions at the same time to help in answering other questions. The user can change his inputs as much as he likes until the entire form seems consistent. When the user has completed a data form, ISAS runs a number of constraint checks to verify that the inputs make sense. The interface itself however, does some simple type checking as soon as the user moves the cursor out of an input field. Such checks include insuring that only digits are typed into a numeric field and that an integer doesn't have a decimal point.

The bottom two lines of every data form are reserved for error messages. One of these lines is used for data type errors that are posted directly by the interface. The other line is for messages generated by the applications program in running its set of constraint checks. Whenever possible, the user is allowed to defer an error, leave the form to perform some other task within the function he is currently in, and return later to correct the error. Error messages are displayed in reverse video for emphasis and whenever possible they are accompanied by positioning the cursor in the field that caused the error.

The interface does have some weak points. In its current implementation, it is very slow: there are noticeable delays in just moving the cursor from field to field on the screen. This is a curable problem, resulting from the choice of languages and data base systems. We chose powerful systems, which allowed us to do faster prototype work, but which were not efficient. This brings us to the discussion of the interface implementation and internals.

### 3.2.2 Implementation of the Man-Machine Interface

The interface implementation is rather complex. The displays are actually managed by a process separate from the one running ISAS's problem solving software. The Unix facilities for spawning new processes and interprocess communication through pipes allow this arrangement to work. The interface manager is written in LISP, and makes use of a relational data base called TROLL and a screen management package called Curses. Communication between the problem solving software, written in FORTRAN, and the interface, is handled by a set of buffer routines written in C.

The motivation for this interface design is also complex. The interface is being developed at AI&DS as a generic interface to be shared by a number of projects, rather than an interface specific to the ISAS needs. ISAS does not fully utilize the capabilities intended in the design. The interface is set up as a separate process so that, potentially, several application programs could use its facilities simultaneously. Each client process is free to run, speaking to the interface only when it needs to, and able to continue its work while the interface takes care of the I/O. It was intended that the interface be able to handle more of the constraint checking on the data input by the user. As noted before, this implementation only does simple type checking on inputs.

The data base is included in the design as the medium of communication between interface and application program. We wanted to take advantage of the relational structure of a data base and its ability to

protect data from loss. It provides a convenient place to store the data for forms and menus, allowing the user to skip from form to form, while keeping previous data available. The data base protects the user from loss of data even in the event of program error. It should also provide a clean and uniform connection to a variety of applications programs.

Most of the efficiency problems alluded to earlier derive from inefficiencies in the individual pieces of the interface. In particular, TROLL turns out to be very slow. Franz Lisp, (the LISP variant available on our VAX) while a good language for development, is not well suited to production programs. The pipe mechanism for inter-process communication is also slow.

We were forced into several modifications for efficiency reasons. Most significant was the decision to pre-assemble the forms and menus needed for ISAS out of the data in TROLL. Once assembled into a screen image, and stored, all changes to the data in TROLL must be put into the appropriate screen image. When called up for display, the image is only updated based on these changes, rather than being reconstructed. A great deal of memory is required to store all the screen images, but the time needed to query the data base in building the images from scratch whenever needed is unacceptable in an interactive system.

The buffer routines, which allow the application program to talk to the interface, are trivial. For the most part, they simply format calls to the LISP interface routines, and send them down the pipe. They make use of C's simple formatted output commands, and are linked in with ISAS's FORTRAN problem solving code.

#### 4. A SURVEILLANCE AVOIDANCE EXAMPLE

This section steps through a typical ISAS session. The displays seen and the actions taken by the user as a problem is specified and solved are described.

##### 4.1 INTRODUCTION TO ISAS

The first forms encountered by an ISAS user provide introductory information. Upon entering the system, the user is presented with a title page, shown in Fig. 4-1. When the user presses the ENTER key as requested on the form, he is taken to the beginning of a tutorial on the use of the man-machine interface. The first page of the tutorial is seen in Fig. 4-2. A user familiar with the use of the menus and data forms will choose to "Proceed to Next Task", while the inexperienced user will request the second page of the tutorial, which is shown in Fig. 4-3. One would also expect the inexperienced user to request general instructions on the operation of ISAS. This is done by moving the cursor to the entry marked "General Instructions" and pressing the ENTER key. Figures 4-4 through 4-7 show the information that is available on the use of the system. At this point, both the experienced and the inexperienced user have completed the introductory section of ISAS and are ready to begin the specification of the surveillance avoidance problem of interest. The three description tasks are MAP SELECTION, SPECIFICATION, and SENSOR SCENARIO.

##### 4.2 PROBLEM SPECIFICATION

The first part of the problem specification is the selection of the map on which the problem is to be solved. As was seen in Fig. 3-2, the user is presented with a simple menu from which he selects a map. In this example, he chooses the "Indian Ocean" option and is shown the map in Fig. 3-4 to ensure that this map is acceptable. The user then requests to "Proceed to Next Task".



INTERACTIVE  
SURVEILLANCE AVOIDANCE SYSTEM  
(ISAS)

Developed By :  
Advanced Information & Decision Systems  
201 San Antonio Circle Suite 206  
Mountain View, CA  
6/6/83

\_(Press "ENTER" to begin the program)

Figure 4-1: ISAS Title Page

- General Instructions                      - Exit ISAS  
- Proceed to Next Task

## USING THE ISAS MENU INPUT SYSTEM

All user input to ISAS is accomplished through menus and forms displayed on this terminal. These are pages of text with designated fields for user input. This current screen is a simple example; the three functions which may be performed at this time are preceded by under-scores.

Using the cursor control keys on this terminal's keyboard, you may move the cursor around the screen, from field to field. These keys have arrows on them: each of these keys moves the cursor in the direction that the arrow points. They are located on the small keypad on the right side of the full keyboard.

In general, to select an option from a menu, you use the cursor control keys to move the cursor to the field next to your choice, and then press "ENTER" to register your decision. The "ENTER" key is located on the same small keypad as the cursor control keys.

**\_ Move the cursor to this position and press "ENTER" to continue reading instructions.**

\_ General Instructions

\_ Exit ISAS  
\_ Proceed to Next Task

---

USING THE ISAS MENU INPUT SYSTEM  
(Continued)

---

To enter data into a form, you similarly move the cursor to the desired field, but then you type in a number or some text, as if filling out a paper form. ISAS will warn you if a value you have tried to enter is unacceptable.

There are two special keys which are useful when filling out forms. They are "DEL CHAR" and "CLEAR REST". "DEL CHAR" will delete the character above the cursor. "CLEAR REST" will clear the rest of the input box starting at the current cursor location. There are three more special keys: "ADD", "DELETE", and "HELP". There are places in ISAS where it is appropriate to use "ADD" or "DELETE" in place of "ENTER" when registering a menu choice. Menus where this occurs will say so, and will explain the effect of using either key. For now, simply note the position of these keys on the same keypad as "ENTER".

At any point in the program, if you do not understand what you are supposed to be doing, you may press the "HELP" key, also located on the small keypad. In response, ISAS will display a page of explanatory text, similar to this page, but specifically tailored to your current location in the program. The option of getting general instructions on the use of ISAS is also available on most pages.

Figure 4-3: Page 2 of Tutorial

\_ Return To Previous Location in ISAS

\_ Exit ISAS

---

GENERAL INSTRUCTIONS  
(Position Cursor: Press "ENTER")

ISAS is an interactive system to aid in planning naval transits that avoid detection by enemy surveillance systems. You must first specify the problem of interest by choosing options from menus and entering information onto computerized data forms. You should then request that the path optimization algorithm be executed (the capability to specify your own path has not been implemented yet). The algorithm will generate one or more paths, any of which can then be examined by requesting to see the results of path evaluation. Each of the major ISAS functions will now be summarized.

CHOOSE A MAP presents a list of available maps to you. Choose the map that best fits the transit to be planned. Your selection will be displayed on the graphics monitor so that you can view several maps in choosing the best one for your problem of interest.

In SPECIFY A TRANSIT PROBLEM you will be asked to provide most of the information that characterizes the problem to be solved. There are four subfunctions that can be executed in any order you wish: DESCRIBE BASIC PROBLEM is a required task that elicits the basic description of the transit; SET QUANTIZATION is also required and asks you to provide

\_ Move to next page of general instructions

Figure 4-4: Page 1 of General Instructions

\_ Return To Previous Location In ISAS

\_ Exit ISAS

---

GENERAL INSTRUCTIONS  
(Position Cursor; Press "ENTER")

some guidance on how to quantise time and space for the path optimisation algorithm; ENTER RENDEZVOUS POINT is an optional task that allows you to specify a rendezvous point and time anywhere along the transit; and ENTER EXCLUSION AREAS is an optional task that allows you to specify some areas that are to be avoided during certain time intervals.

A sensor scenario will be automatically read into the system. The scenario is characterized by the list of enemy sensor platforms relevant to the area of the world and time interval of interest; the type, position, motion model, and list of sensors on each platform, and the model of detection probability for each sensor. In EXAMINE/EDIT THE SENSOR SCENARIO you are presented a list of relevant platforms. You may delete any platform, ask to examine/edit any platform, or add a new platform. Either of the latter two options takes you to a data form specific to the type of platform to be added or edited. You can then edit position and motion information for the platform of interest, add or delete sensors from the platform, or ask to examine the model for any sensor appropriate to the platform type.

\_ Move to next page of general instructions

Figure 4-5: Page 2 of General Instructions

\_ Return To Previous Location In ISAS

\_ Exit ISAS

---

GENERAL INSTRUCTIONS  
(Position Cursor: Press "ENTER")

When you finish editing the sensor scenario, the surveillance avoidance problem is completely specified. You then are given a choice between RUN THE PATH OPTIMIZATION ALGORITHM and ENTER AND EVALUATE USER CREATED PATH. The second option would allow you to examine the sensor coverage as it evolves with time and to use this information in creating your own surveillance avoidance path. This capability has not yet been implemented so you should choose the algorithm option instead.

RUN THE PATH OPTIMIZATION ALGORITHM executes a dynamic programming algorithm to generate an optimal solution to the surveillance avoidance problem. The algorithm may generate several paths while searching for the optimum. Each of these paths will be displayed to you after it is generated.

To examine in greater detail any of the paths created by the algorithm, select SHOW PATH EVALUATION RESULTS from the menu. Within this function, TRANSIT DISPLAY (GRAPHICAL) allows you to view the map, the path of interest, and the position-dependent detection probabilities over the problem area for any time of interest during the transit, and presents textual information on the evading ship's current position, heading, speed of advance, fuel use, probability of being detected, and whether or not line-of-sight emissions are safe. TRANSIT SUMMARY

\_ Move to next page of general instructions

\_ Return To Previous Location In ISAS

\_ Exit ISAS

---

GENERAL INSTRUCTIONS  
(Position Cursor; Press "ENTER")

(TABULAR-HARD COPY) provides a hard copy summary of the textual information corresponding to every time interval during the transit. GRAPH OF FUEL CONSUMPTION VS. DETECTION has not been implemented yet. This function would generate and display a graph plotting fuel consumption versus probability of detection for each of the available paths to allow you to choose the most desirable path.

At most points within ISAS you are given four options in a menu found at the top of the screen. GENERAL INSTRUCTIONS prints the information you are reading now. EXIT ISAS allows you to exit the system. RE-SPECIFY PROBLEM allows you to return to any previously completed problem specification function (CHOOSE A MAP, SPECIFY A TRANSIT PROBLEM, EXAMINE/EDIT THE SENSOR SCENARIO) to modify the problem of interest. PROCEED TO NEXT TASK will either take you automatically to the next top-level function to be performed or will present you with a menu of options to choose from.

Figure 4-7: Page 4 of General Instructions

The next description task contains four subtasks and is the largest and most important of the description tasks. The four subtasks are seen as options on the menu first presented in Fig. 3-5. The user selects the option marked "Describe Basic Problem" and begins inputting the basic information requested. Figure 4-8 shows the completed "Describe Basic Problem" data form for this example. The user has requested a nine-day transit of an aircraft carrier from the Strait of Malacca to a point a few hundred nautical miles outside of the Gulf of Oman, using no more than 25% of the fuel capacity of the ship. Having completed this form, the user chooses to perform "Set Quantization" next. He requests a seven-hour decision interval, maximum starboard and port deviations of 300 and 1000 n. miles, respectively, and the coarse resolution option, as seen in Fig. 4-9. The user has now completed the two required subtasks in this section of ISAS, but chooses to input several exclusion areas. The inputs for these exclusion areas are shown in Fig. 4-10. The user chooses not to request a rendezvous point, so selects "Proceed to Next Task" at this time.

The next specification task is the examination and possible editing of the sensor scenario. The form shown in Fig. 4-11 contains a list of relevant sensor platforms for the transit. A pictorial summary of the sensor scenario (this summary is not an ISAS capability) is given in Fig. 4-12. The scenario has two fixed sensors: a radar located at the southern tip of Sri Lanka and a SOSUS platform just off the coast of Somalia. There are two enemy ships patrolling barrier search patterns. There is also one enemy ship transiting from  $20^{\circ}$  n,  $62^{\circ}$  e to  $8^{\circ}$  n,  $95^{\circ}$  e, and one transiting from  $1^{\circ}$  s,  $85^{\circ}$  e to  $24^{\circ}$  n,  $61^{\circ}$  e. One RORSAT satellite with an inclination of  $62^{\circ}$  and a period of 88.3 minutes is also included in the scenario.

In this example the user has no new information regarding enemy sensor platforms and does not wish to examine the scenario in greater detail, so he selects the "Proceed to Next Task" option, thus completing the problem specification portion of ISAS.



Surveillance Avoidance  
A Surveillance Avoidance Example

Final Report  
Section 4

<input type="checkbox"/> General Instructions	<input type="checkbox"/> Exit ISAS
<input type="checkbox"/> Re-Specify Problem	<input type="checkbox"/> Proceed to Next Task

---

SPECIFY A TRANSIT PROBLEM  
(Position Cursor: Press "ENTER")

<input checked="" type="checkbox"/> Describe Basic Problem (Required)	<input type="checkbox"/> Set Quantization (Required)
<input type="checkbox"/> Enter Rendezvous Point (Optional)	<input type="checkbox"/> Enter Exclusion Areas (Optional)

---

Describe Basic Problem (Required)  
(Position Cursor: Enter Data)

Transit	Latitude	Longitude	Date	Time (GMT)
Start	7 0 0 n	97 0 0 e	7 / 24 / 83	1200
finish	22 0 0 n	64 30 0 e	8 / 4 / 83	1730

Describe Your Ship -

Maximum Speed (knots)	30
Minimum Speed (knots)	5
Maximum Allowed Fuel Consumption (% Fuel Capacity)	25

Ship Type (Position Cursor: Press ENTER)

<input type="checkbox"/> Cruiser
<input type="checkbox"/> Destroyer
<input checked="" type="checkbox"/> Aircraft Carrier

Figure 4-8: Completed Describe Basic Problem Form

<input type="checkbox"/> General Instructions	<input type="checkbox"/> Exit ISAS
<input type="checkbox"/> Re-Specify Problem	<input type="checkbox"/> Proceed to Next Task

---

SPECIFY A TRANSIT PROBLEM  
(Position Cursor, Press "ENTER")

---

<input type="checkbox"/> Describe Basic Problem (Required)	<input checked="" type="checkbox"/> Set Quantization (Required)
<input type="checkbox"/> Enter Rendezvous Point (Optional)	<input type="checkbox"/> Enter Exclusion Areas (Optional)

---

Set Quantization  
(Position Cursor, Enter Data)

Approximate Interval Between Course Changes (Hours) :	2
Maximum Deviation From Mean Track	
to Starboard (N. Miles)	300
To Port (N. Miles)	1000

Precision Option (Choose One, Position Cursor, Press "ENTER")

<input checked="" type="checkbox"/> Low Resolution Solution / Short Run-Time
<input type="checkbox"/> Medium Resolution Solution / Medium Run-Time
<input type="checkbox"/> High Resolution Solution / Long Run-Time

Figure 4-9: Completed Set Quantization Form

AD-A196 253

SURVEILLANCE AVOIDANCE TECHNIQUE DEMONSTRATION(U)  
ADVANCED INFORMATION AND DECISION SYSTEMS MOUNTAIN VIEW  
CA J M ABRAM ET AL. 02 DEC 83 AT/DS-TR-1026/10-1

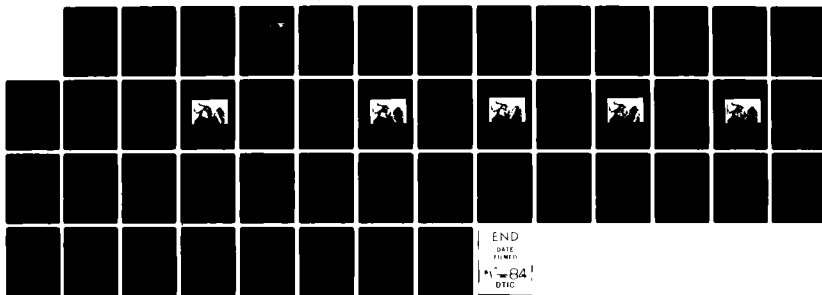
2/2

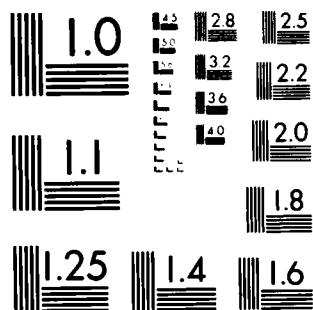
UNCLASSIFIED

N00014-82-C-0085

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

- General Instructions
- Re-Specify Problem
- Exit ISAS
- Proceed to Next Task

**SPECIFY A TRANSIT PROBLEM**  
(Position Cursor: Press "ENTER")

- Describe Basic Problem (Required)
- Enter Rendezvous Point (Optional)
- Set Quantization (Required)
- Enter Exclusion Areas (Optional)

**ENTER EXCLUSION AREAS**

Enter up to ten circular exclusion areas and time intervals

	Center		Size (N. Miles)		From		Until	
	Latitude	Longitude	Radius		Date	Time(CMT)	Date	Time(CMT)
1	2 0 0 n	43 0 0 e	180		7 /26/83	1700	8 7 83	1200
2	10 0 0 n	70 0 0 e	150		7 /27/83	1400	8 5 83	0700
3	16 0 0 n	72 0 0 e	200		7 /28/83	1800	8 7 83	2100
4	7 0 0 n	77 0 0 e	120		7 /26/83	0800	8 3 83	1500
5					/ /			
6					/ /			
7					/ /			
8					/ /			
9					/ /			
10					/ /			

Figure 4-10: Completed Enter Exclusion Areas Form

General Instructions

Re-Specify Problem

Exit ISAS

Proceed to Next Task

---

EXAMINE / EDIT THE SENSOR SCENARIO

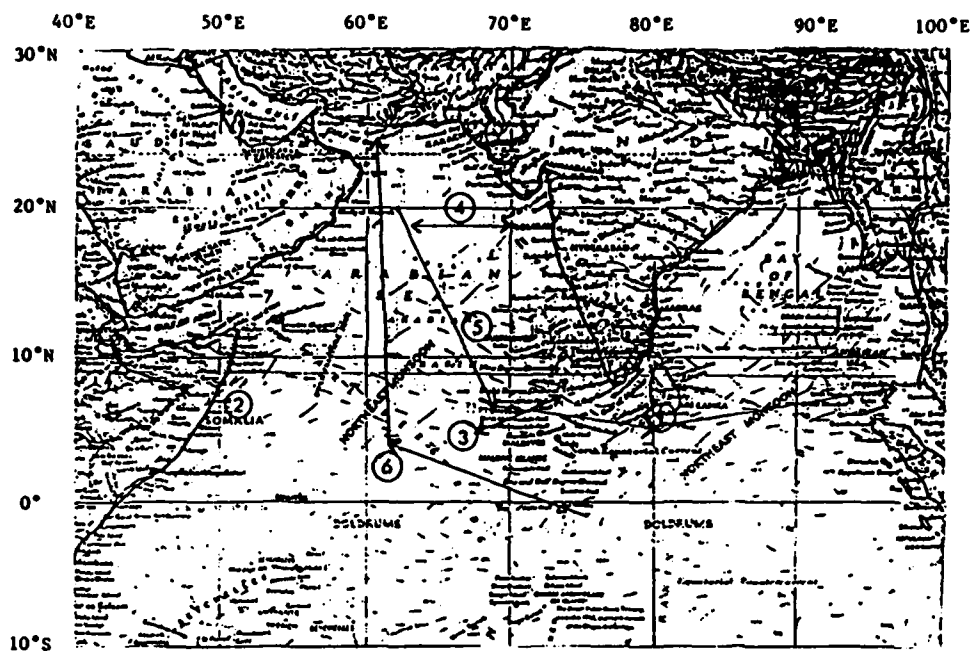
(Position Cursor: Press "ENTER" to examine/edit platform:  
Press "ADD" to create new platform; Press "REMOVE" to delete platform)

Platform Name	Platform Type
New	Earth-Based, Fixed
New	Earth-Based, Barrier Search
New	Earth-Based, Transiting
New	Satellite

---

Ceylon Radar	Fixed, Earth Based
Somalia SOSUS	Fixed, Earth Based
Surveillance Ship 1	Barrier Search
Surveillance Ship 2	Barrier Search
Transiting Ship 1	Transitting
Transiting Ship 2	Transitting
Satellite 1	Satellite

Figure 4-11: List of Platforms for Example



<u>Surface Platforms</u>		<u>Satellite</u>
<u>Platform #</u>	<u>Sensor Types</u>	Sensor: ROR
1	Search Radar	Period: 88.3 Min.
2	SOSUS	Inclination: 62°
3	Search Radar, ESM	
4	Search Radar, TASS	
5	Search Radar, ESM	
6	Search Radar, FSM	

Figure 4-12: Pictorial Summary of Sensor Scenario

#### 4.3 PATH OPTIMIZATION

Upon proceeding to the next task from the SENSOR SCENARIO function, the user is presented with the selection menu depicted in Fig. 4-13. Since the option to "Enter and Evaluate User Created Path" is not implemented at this time, the user chooses to "Run the Path Optimization Algorithm". The form that was shown in Fig. 3-38 informs the user that the path optimization process has begun. Figures 4-14 through 4-19 show the forms that summarize the six paths generated during the path optimization process as the dynamic programming algorithm sought the optimal solution satisfying the fuel constraint. Note that path1 is the optimal unconstrained solution and has lower detection probability and higher fuel use than any other path. Since the unconstrained optimum does not satisfy the fuel constraint of 25% of capacity, a Lagrange multiplier is chosen and another DP run is required. The second path is the same as the first because the Lagrange multiplier was too small, so a larger multiplier is chosen and another run is made. Path3 nearly satisfies the fuel constraint, requiring 25.88% of capacity, and has a higher detection probability than the first two paths, as expected. The fourth path is the first to satisfy the fuel constraint, but has detection probability almost double that of the first two paths. The final two paths are the same as path4 and were generated by relaxing the Lagrange multiplier in the hopes of finding a feasible path with lower detection probability than the .659 yielded by path4. No such path can be found, so the algorithm concludes that path6 is the optimum, and the user is presented with the form shown in Fig. 4-20. The user is now ready to proceed to the next task, and indicates this by making the corresponding menu selection.

#### 4.4 EXAMINING THE RESULTS

In principle, the user has the choice of examining the results of the path optimization algorithm or creating a path of his own upon completion of the algorithm through use of the menu shown in Fig. 4-21. By default, the user chooses the first of these options because the second has yet to be implemented. The menu that was shown in Fig. 3-43 for



\_ General Instructions

\_ Exit ISAS

---

PROCEED TO NEXT TASK

\_ Run the Path Optimization Algorithm

\_ Enter and Evaluate User Created Path

Figure 4-13: OPTPATH Menu

Run the Path Optimization Algorithm  
(Wait for prompt; keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name :	path1	
Cumulative Probability of Detection :	0.377	
Cumulative Fuel Use (% Capacity) :	39.51	

Figure 4-14: Path 1 Brief Summary

Run the Path Optimisation Algorithm  
(Wait for prompt, keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name :	path2
Cumulative Probability of Detection :	0.377
Cumulative Fuel Use (% Capacity) :	39.51

Figure 4-15: Path 2 Brief Summary

Run the Path Optimization Algorithm  
(Wait for prompt; keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name : path3  
Cumulative Probability of Detection : 0.491  
Cumulative Fuel Use (% Capacity) : 25.88

Figure 4-16: Path 3 Brief Summary

Run the Path Optimization Algorithm  
(Wait for prompt; keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name :	path4	
Cumulative Probability of Detection		0.659
Cumulative Fuel Use (% Capacity)		28.56

Figure 4-17: Path 4 Brief Summary

Run the Path Optimization Algorithm  
(Wait for prompt; keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name :	path5
Cumulative Probability of Detection	0.459
Cumulative Fuel Use (% Capacity)	20.56

Figure 4-18: Path 5 Brief Summary

Run the Path Optimisation Algorithm  
(Wait for prompt; keyboard will not respond)

A path has been generated and evaluated. It may not be optimal.

Path Name :	path6
Cumulative Probability of Detection :	0.639
Cumulative Fuel Use (% Capacity) :	20.56

Figure 4-19: Path 6 Brief Summary

- General Instructions
- Re-Specify Problem
- Exit ISAS
- Proceed to Next Task

---

Run the Path Optimization Algorithm  
(Position Cursor; Press "ENTER")

The path optimization algorithm has been completed. path6  
has been determined to be the optimal path and has been  
renamed "optimum". Please choose an item from the menu above.

Figure 4-20: End of Optimization for Example



\_ General Instructions

\_ Exit ISAS

---

PROCEED TO NEXT TASK

\_ Enter and Evaluate User Created Path

\_ Show Path Evaluation Results

Figure 4-21: SHOWPATH Menu

examining the results has three selections. Our user chooses "Transit Summary" and is then given a menu of paths to select from, as seen in Fig. 4-22. The user selects "OPTIMUM" and receives the printout that was displayed in Fig. 3-48. The user now wishes to examine the optimal path graphically, and does so by selecting "Transit Display" from the menu. Again, a path menu is presented, as was seen in Fig. 3-44, the user selects "OPTIMUM" from the menu, and the user commences his examination of the graphical displays. Figures 3-45 and 3-46 showed the graphics monitor display and the corresponding display of tabular information on the Ann Arbor terminal for the first leg of the transit.

At the beginning of the first leg, the evading ship is at its starting location at the northeast end of the Strait of Malacca. The entire transit path is displayed, and the double line extending from the current location of the evading ship shows the distance and direction to be traveled during the first leg. The large blue area, darker than the surrounding ocean and between the Gulf of Aden and India, is the detection probability arising from the combination of the SOSUS sensor located off the Somalian coast and a satellite footprint. the location of the SOSUS itself is not shaded because the problem grid does not extend that far to the west. The satellite footprint can be seen passing through the SOSUS coverage region in a northeasterly direction. A small radar coverage area is seen at the southern tip of Sri Lanka. The first barrier search platform is represented by the four shaded cells southeast of the square denoting the destination and the second barrier search platform is shown by two shaded cells west of the southern tip of India. The first transiting enemy ship is south of the evading ship's destination and the second is south of Sri Lanka. The two black regions are the two exclusion areas that are active at the beginning of the transit.

When finished with the examination of stage 1 of the transit, the user requests "Display Next Stage". Figures 4-23 and 4-24 show the situation for stage 2. Notice that the positions of all of the ships have changed since the previous display and that the satellite is now moving to the southeast. There has been very little detection threat in

- General Instructions  
- Re-Specify Problem

- Exit ISAS  
- Proceed to Next Test

---

SHOW PATH EVALUATION RESULTS  
(Position Cursor: Press "ENTER")

- Transit Display (Graphical)  
- ☒ Transit Summary (Tabular - Hard Copy)  
- Graph of Fuel Consumption vs Detection

---

TRANSIT SUMMARY (Tabular - Hard Copy)  
(Position Cursor: Press "ENTER")

Path Name

\_path1  
\_path2  
\_path3  
\_path4  
\_path5  
\_OPTIMUM

Figure 4-22: TABULAR Path Menu



Figure 4-23: Graphical Display for Stage 2

General Instructions      Exit ISAS  
Path Examination Complete

TRANSIT DISPLAY (GRAPHICAL)

Tabular Information for Graphical Display of Current Stage

Path Name OPTIMUM

Stage Length: 7.145

STAGE	DATE	TIME	POSITION	HEADING	SPEED OF	CUM.	DETECTION	LOS
0		(GMT)	LAT LONG	(DEG)	ADV	FUEL	PROBABILITY	EMIS
					(KNOTS)	(% CAP.)	CUM. NEXT	SAFE?
2	7 26 83	1909	43n 94 62e	208.68	14.55	1.969	0.000 0.000	YES

STEP THROUGH STAGES OF TRANSIT  
(Position Cursor: Press ENTER)

Display Previous Stage

Display Next Stage

JUMP TO ANY STAGE OF TRANSIT  
(Enter Data: Press ENTER)

Display Stage Number

or  
Display Stage Containing - Date: / / Time (GMT).

Figure 4-24: Tabular Information for Stage 2

this early part of the transit.

The user now requests a jump to stage 9, that situation being displayed in Figs. 4-25 and 4-26. Notice that the satellite footprint is near the evading ship and that the dynamic programming algorithm chose a speed and direction for the next transition that kept the ship outside of the satellite coverage area. Also, all four of the exclusion areas are now active and can be seen in the graphical display.

Jumping to stage 11, as depicted in Figs. 4-27 and 4-28, the user examines the first instance of detection probability. As seen in Fig. 4-27, approximately half of the transition for stage 11 is within the satellite coverage region. The probability of detection arising from the exposure to the satellite is .182. Most likely this detection is necessary in order to satisfy the fuel constraint without taking a higher probability of detection later in the transit as a result of avoiding detection here.

The user now jumps to stage 17, seen in Figs. 4-29 and 4-30. The entire transition for stage 17 lies within a satellite footprint, yielding a detection probability of .331 for the transition and a cumulative detection probability of .453 at the end of the stage. However, it can be seen that detection by the second barrier search platform is avoided by passing just south of that platform's coverage area.

The user now jumps to the final stage, shown in Figs. 4-31 and 4-32. In this stage there is an unavoidable detection by the satellite because the destination lies within the satellite footprint. The much higher detection probability arising from the first barrier search platform is avoided, however. Total detection probability for the transit is .659 and fuel consumption is 20.56% of capacity.

Several options are available to the user at this point. He could examine another path, such as path3, which yielded detection probability of .491 while only slightly exceeding the fuel constraint. He could ask for a tabular summary of another path. Or he could choose to respecify



Figure 4-25: Graphical Display for Stage 9

## Final Report Section 4



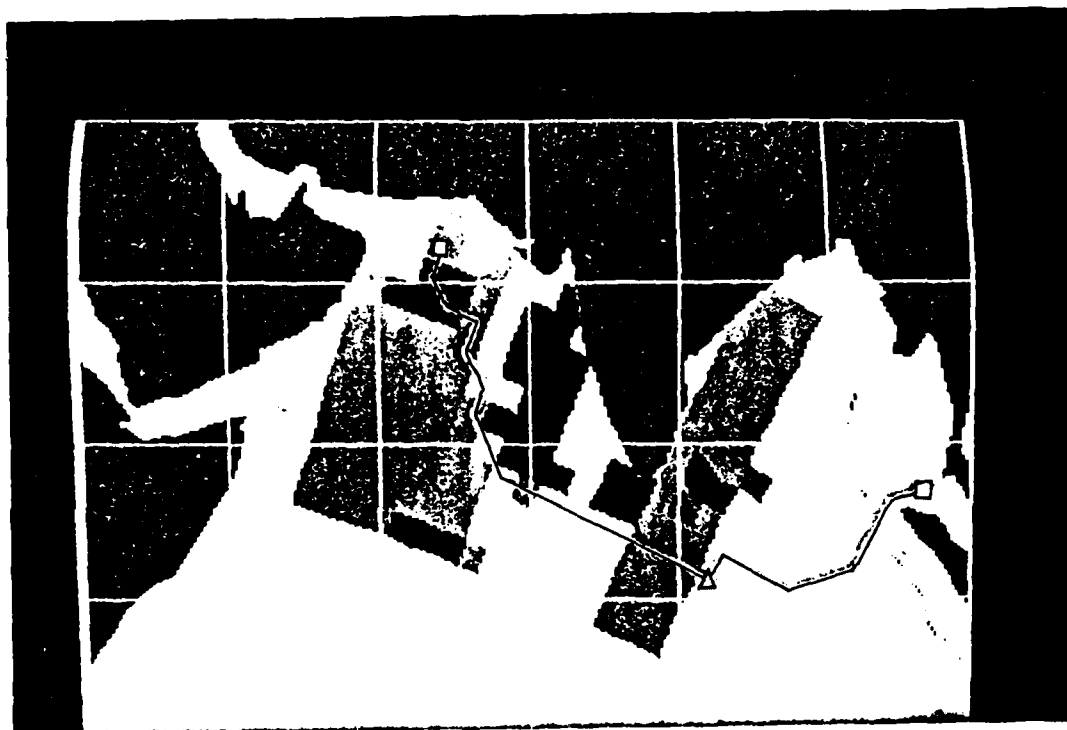


Figure 4-27: Graphical Display for Stage 11

Surveillance Avoidance  
A Surveillance Avoidance Example

Final Report  
Section 4

\_ General Instructions

\_ Exit ISAS

\_ Path Examination Complete

TRANSIT DISPLAY (GRAPHICAL)

Tabular Information for Graphical Display of Current Stage

Path Name: OPTIMUM

Stage Length: 7.145

STAGE	DATE	TIME (GMT)	POSITION LAT LONG	HEADING (DEG)	SPEED OF ADV (KNOTS)	CUM. FUEL (% CAP.)	DETECTION PROBABILITY CUM. NEXT	LOS EMIS SAFE?
11	7 29 83	1127	1.1n 82.11e	296.65	14.23	0.482	0.182 0.182	YES

STEP THROUGH STAGES OF TRANSIT  
(Position Cursor: Press ENTER)

\_ Display Previous Stage

\_ Display Next Stage

JUMP TO ANY STAGE OF TRANSIT  
(Enter Data: Press ENTER)

Display Stage Number

or

Display Stage Containing - Date: / / Time (GMT):

Figure 4-28: Tabular Information for Stage 11

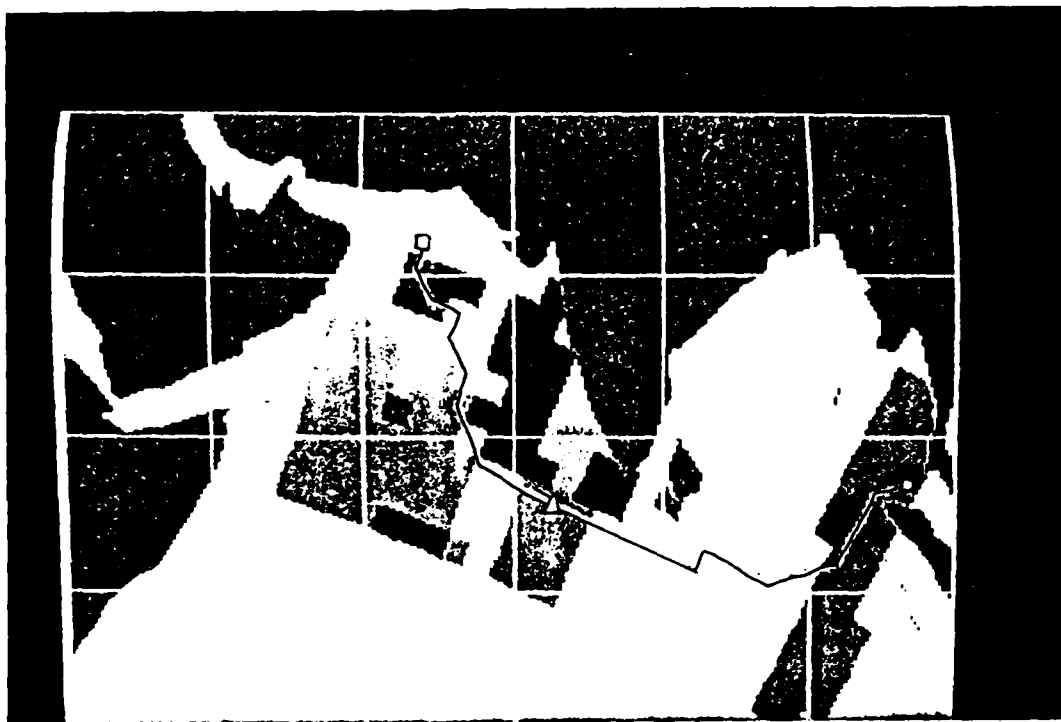


Figure 4-29: Graphical Display for Stage 17

Surveillance Avoidance  
A Surveillance Avoidance Example

Final Report  
Section 4

\_ General Instructions

\_ Exit ISAS

\_ Path Examination Complete

TRANSIT DISPLAY (GRAPHICAL)

Tabular Information for Graphical Display of Current Stage

Path Name: OPTIMUM

Stage Length: 7.145

STAGE #	DATE	TIME (GMT)	POSITION LAT LONG	HEADING (DEG)	SPEED OF ADV (KNOTS)	CUM. FUEL (% CAP.)	DETECTION PROBABILITY CUM. NEXT	LOS ENIS SAFE?
17	7 31 83	0619	5.5n 72.90e	293.84	14 23	11.56	0.453 0.331	YES

STEP THROUGH STAGES OF TRANSIT  
(Position Cursor: Press ENTER)

\_ Display Previous Stage

\_ Display Next Stage

JUMP TO ANY STAGE OF TRANSIT  
(Enter Data: Press ENTER)

Display Stage Number

or

Display Stage Containing - Date: / / Time (GMT)

Figure 4-30: Tabular Information for Stage 17



Figure 4-31: Graphical Display for Stage 31

Surveillance Avoidance  
A Surveillance Avoidance Example

Final Report  
Section 4

General Instructions                      Exit ISAS  
   Path Examination Complete

TRANSIT DISPLAY (GRAPHICAL)

Tabular Information for Graphical Display of Current Stage

Path Name: OPTIMUM

Stage Length: 7.145

STAGE	DATE	TIME	POSITION	HEADING	SPEED OF	CUM.	DETECTION	LOS
9		(GMT)	LAT LONG	(DEC)	ADV	FUEL	PROBABILITY	EMIS
					(KNOTS)	(% CAP.)	CUM. NEXT	SAFE?
31	8 4 83	1021	22.n 64.49e	0.0000	0.000	20.54	0.459 0.331	YES

STEP THROUGH STAGES OF TRANSIT  
(Position Cursor: Press ENTER)

Display Previous Stage

Display Next Stage

JUMP TO ANY STAGE OF TRANSIT  
(Enter Data: Press ENTER)

Display Stage Number :  
or

Display Stage Containing - Date: / / Time (GMT):

Figure 4-32: Tabular Information for Stage 31

the problem, perhaps increasing the fuel constraint or changing the destination position or date/time to avoid the final satellite detection. This is the action taken by the user in this example, the menu of Fig. 4-33 is displayed to the user, and the user can respecify the problem as desired and generate another optimal solution.

\_ General Instructions

\_ Exit ISAS

---

RE-SPECIFY PROBLEM  
CURRENTLY AT: SHOW PATH EVALUATION RESULTS  
(Position Cursor: Press "ENTER")

\_ Choose a Map

\_ Specify a Transit Problem

\_ Examine / Edit the Sensor Scenario

\_ Return to: Show Path Optimisation Results

Figure 4-33: Respecify Menu After SHOW RESULTS



## 5. ENGINEERING ASSESSMENT

In this section the memory and runtime requirements of the current ISAS implementation are discussed, followed by an engineering assessment of a subsequent operational ISAS implementation.

### 5.1 CURRENT MEMORY AND RUNTIME REQUIREMENTS

The current version of ISAS is a development-oriented implementation designed to prove feasibility of concept. Its memory and runtime needs are therefore much greater than what could be achieved through additional refinement aimed at streamlining the system. Storage space poses few problems on the VAX 11/750 with its virtual memory system, 2 Mbytes of core memory, and 330 Mbytes of disk memory. Optimal storage of the data required by ISAS was not a driving concern in developing the demonstration version of the software.

The declaration sizes for many of the arrays used by ISAS are determined by the upper bounds established for the number of states and stages allowed for the dynamic programming (DP) problem. The DP algorithm currently allows a 50x50 state space and 50 stages.

#### 5.1.1 Memory Requirements

The total compiled ISAS software requires 270,000 bytes of memory; 200,000 is for the applications program and 70,000 is the man-machine interface software. The uncompiled source code for ISAS takes up 305,000 bytes of storage. In addition, 67,000 bytes are used for data files, 61,700 of which is needed to store the 240x255 pixels map. Forms needed by the man-machine interface take another 175,00 bytes. There is also currently 1,800,000 bytes of memory allocated to declared arrays and variables used by ISAS. The total memory requirement for ISAS is therefore approximately 2.7 Mbytes.

Ideally, the current implementation of ISAS should be run on a system that can provide at least 300 to 400 K of core memory so that the compiled program can fit entirely within the core, with some space remaining for the data that is being acted on at any given time. Hardware with less available core can be used only if it can bring in pieces of compiled program from disk as needed. The remainder of the current ISAS memory requirements could fairly easily be decomposed into smaller pieces, stored offline, and read into core as needed.

#### 5.1.2 Runtime Requirements

A typical complete ISAS session requires approximately 18 minutes of CPU time on the VAX 11/750. It should be pointed out that the VAX is a flexible general purpose machine that is fairly inefficient in terms of simple number crunching power. The VAX has no floating point hardware, meaning that floating point calculations, which are avoided when possible in the current implementation, are extremely slow. Runtime requirements are also inflated by the inefficiency of the experimental man-machine interface used in ISAS. The CPU time requirements are now discussed in greater detail.

From the start of an ISAS session to the beginning of the SPECIFICATION function takes about 3 CPU minutes. This time is consumed by the initialization procedures, the man-machine interface for introductory forms and for map selection, and for transferring the chosen bit map from the VAX to the graphics monitor via the CROMEMCO. Another 2 CPU minutes are then expended on man-machine interface and local constraint checking for the four subtasks within the SPECIFICATION function. The global constraint checking, grid construction, and generation of cell to pixel mapping requires approximately 2.5 CPU minutes. In the next function, SENSOR SCENARIO, the required CPU time depends primarily on the number of satellites in the scenario. Each satellite currently needs almost 4 minutes of CPU time for calculation of footprints and quantized detection probabilities for the entire transit. The example given in Section 4 had one satellite and used just over 4 minutes of CPU time within SENSOR SCENARIO. Each iteration of the dynamic programming algorithm needs

1.5 CPU minutes. Estimating the average number of required iterations to be 4, optimal path generation requires 6 minutes of CPU time. Another CPU minute is typically used within SHOW RESULTS, bringing the total time of a typical ISAS session to 18.5 CPU minutes.

## 5.2 ENGINEERING ASSESSMENT FOR FUTURE IMPLEMENTATION

As mentioned several times already, ISAS was developed to demonstrate feasibility of technique and would require significant further development for it to become a useful operational system. Section 6.1 includes a discussion of the further efforts that would be required before installing ISAS in an operational environment. Improved efficiency would be one goal of a further development effort. Both memory and runtime requirements could be reduced significantly by expending efforts to improve the efficiency of the ISAS implementation. There are certain tradeoffs between memory and runtime in the system so the amount of reduction possible in the two areas depends critically on the capabilities and needs of the target hardware and user. In this section, the memory and runtime needs of a streamlined ISAS are assessed.

### 5.2.1 Assessment of Memory Needs

With more attention paid to programming efficiency and with a more efficient compiler, it is estimated that the ISAS code could be enhanced for an operational environment while reducing the size of the compiled software slightly to approximately 250,000 bytes. The size of the source code for this future implementation would probably increase somewhat, with an estimated requirement of 325,000 bytes. It is assumed that sensor scenario information and a library of maps would be available from existing data bases distinct from but accessible by ISAS, so that no data file storage space would be required by the system. Information could go directly from the existing data bases to the appropriate arrays in the system software. The system arrays and variables would require approximately 800,000 bytes of memory, under the assumption that floating point hardware would be available, thus eliminating the need for the large integer-valued probability of detection arrays now

required by ISAS. The forms for the man-machine interface would take up approximately 250,000 bytes, bringing the total estimated storage requirement to 1.6 Mbytes.

These are, of course, rough estimates of memory requirements. Exact requirements would depend on the language and compiler used, the number of enhanced capabilities requested by the potential user, and the resolution level required of the paths generated by the system. Storage requirements can be reduced significantly by reducing the upper bounds on the number of allowed states and stages for the dynamic programming solution technique.

#### 5.2.2 Assessment of Runtime Needs

There are several ways in which the runtime of ISAS could be reduced in a future implementation. The use of hardware that can perform simple mathematical operations at a faster rate than the general purpose VAX 11/750 currently being used could reduce the runtime of certain sections of the software by at least a factor of 2. The man-machine interface, with its combination of FORTRAN, LISP, C, and TROLL relational data base system could more efficiently be written in a single language without losing any of the capabilities required by ISAS. The subroutine used to calculate satellite footprints contains recently discovered inefficiencies. A new subroutine has been designed, but not implemented, that could reduce the average calculation time from 4 CPU minutes to 1 CPU minute per satellite. By implementing these changes, runtime of the system could be reduced to approximately 12 minutes of CPU time for a typical session.

Another technique for reducing runtime is called successive refinements. In this technique one begins by solving the surveillance avoidance problem on a grid much coarser than the one desired. This yields a path that is a candidate for refinement. The path is subsequently refined by discretizing the area surrounding the candidate path, using smaller cells than were used originally. Note that only the area near the original path is searched to yield a refined path, in general a

much smaller area than the full area of interest so that the search for an optimal solution is much faster. The path produced by refinement can be further refined until the desired granularity is reached. The disadvantage of this approach is that the optimal path for a fine discretization of the full area of interest will not be found if it is not close to the optimal paths for coarser discretizations. By implementing this technique, runtime could probably be reduced to approximately 9 minutes per average session. Further reductions in the runtime could be achieved by more severely limiting the maximum allowable problem size, at the cost of reducing the resolution of the solutions generated by the system.

The runtime estimates given above are based on an implementation similar to the current one. As mentioned earlier, there are certain tradeoffs between available memory and runtime that must be investigated if ISAS is to be implemented on a Navy machine. If there is not enough available memory to store all of the detection probability information required by the dynamic programming algorithm, either in core or on disk, runtime of the system will increase because it will be necessary to calculate detection probabilities whenever needed rather than computing them once and storing them for future use. Also, an implementation on hardware with limited core memory but sufficient disk capabilities will run more slowly due to the system overhead required to shift information in and out of core as needed. not close to the optimal paths for coarser discretizations.

## 6. CONCLUSIONS

This section contains a discussion of future efforts that could be spent developing the ISAS surveillance avoidance aid and a summary of work performed on the contract.

### 6.1 POTENTIAL FUTURE EFFORTS

The current ISAS was developed to demonstrate the feasibility of the dynamic programming approach to solving surveillance avoidance problems. As such, it does not have all of the components that would be required in an operational version of the system. Further development would be required before implementing ISAS in an on board or shore-based Navy environment. Some potential future development efforts are now described.

The information describing the sensor scenario and the map of interest currently resides within files that were created in-house for purposes of demonstrating the performance of the system. An operational ISAS would have to interface to existing intelligence and map data bases in order to extract the information necessary for solving the surveillance avoidance problem. The interface to the intelligence data base would have to find all of the sensor platforms relevant to the transit area and time interval, look up information on the position, motion model, name, ship type, and list of sensor types for each platform, and put this information into an appropriate format for ISAS. The interface to the map data base would have to be able to distinguish land masses from bodies of water, and together with the ISAS man-machine interface would have to provide the user with the capability to choose the most appropriate map with respect to region and scale.

The models of platforms and sensors are simplified models that must be improved for ISAS to be used operationally. The platform motion models are currently deterministic. While deterministic models are fine

for enemy satellites, whose orbits rarely change, and for those fixed platforms that don't often move and whose positions are known, deterministic information on the motion of surface or subsurface vessels is rarely available. Stochastic predictions of future positions of these ships can be made based on track history, perceived intent, Soviet naval doctrine, etc. Stochastic motion models should be used for the moving platforms that are known to be in the vicinity of the transit. ISAS can accommodate stochastic models by spreading the probability of detection induced by a moving platform over its probabilistically predicted positions. Also, unknown sensor threats, especially submarines, would be expected in most transit areas. If any information is available on the likelihood of finding unknown threats in various regions, this could be probabilistically included in the ISAS sensor scenario model so that regions likely to contain unknown threats could be avoided.

The sensor models used by ISAS were created in-house and are reasonable though not realistic. Earth-based models consist of a maximum of ten concentric rings with uniform detection probability within each ring. Satellite sensors are described by distance from the orbital plane to beginning of coverage, width of coverage area, and uniform detection probability within the coverage swath. The performance of acoustic sensors is modeled to be independent of the speed of the transiting ship. Realistic models of the performance of the actual sensors that would be encountered on Soviet platforms should be obtained and incorporated into ISAS.

The cost function that ISAS minimizes in generating an optimal path is the cumulative probability that the evading ship is detected during the transit. This function is a reasonable and tractable cost function for the surveillance avoidance problem. However, it is not always the function of most interest to the user. In a fairly dense sensor environment, the cumulative probability of detection may be close to one for every possible path, so that there is little difference between the optimal path and any other path with respect to the current cost function. In this situation there could be significant differences between candidate paths with respect to other cost functions. One alternative

cost function is weighted detection probability: during each part of the transit, the user could specify some value of avoiding detection by enemy sensors, ranging from unimportant to vital. In the dense sensor example where detection somewhere along the transit is almost certain, the user could ask that ISAS especially avoid detection during the most critical part of the transit. In other situations, detection alone may not concern the user. It may be more desirable to minimize the probability that enemy sensors can detect the ship often enough to establish a track or to fire weapons accurately. These and other cost functions should be investigated further.

As described in greater detail in Section 3.1.6, a useful capability that could be added to ISAS is that of allowing the user to input entire paths or to vary portions of existing paths and to then allow ISAS to evaluate the new path. Some additional effort will be required to add this to ISAS.

The performance of certain sensors can be degraded by the use of countermeasures. An obvious example is a line-of-sight ESM that can be defeated by not emitting electromagnetic signals while in its range. Other possible countermeasures include EM spoofing, jamming, decoying, and zig-zagging. ISAS implicitly treats EMCON by assuming that line-of-sight emissions are prohibited during the entire transit so that ESM sensors do not contribute to the probability of detecting the evading ship, but ISAS informs the user whether LOS emissions are safe or not for each leg of the transit. This approach was chosen so that avoidance performance would not be degraded by attempting to avoid the ESM sensors when it is so much easier to merely use EMCON for the short period that the ship is within coverage. A future version of ISAS might consider the costs and benefits of other countermeasures and aid the user in choosing the most appropriate measure to take.

Weather effects could play an important role in the surveillance avoidance problem. Weather can effect the motion models and fuel consumption rates for ships. The user may want to avoid certain weather patterns; these patterns could be specified as moving avoidance areas by



the user. More importantly, certain sensors, particularly satellite-based optical and infrared sensors, have difficulty penetrating cloud cover so that masking effects could be exploited. Since weather is a highly volatile phenomenon, weather effects would be treated stochastically, but there are certainly gains to be made in including weather effects into the ISAS models.

The future efforts proposed thus far relate to adding new capabilities into ISAS. There are also a number of areas in which work could be done to improve the existing capabilities. The entire system could be rewritten in a more suitable language, such as Ada or C. The man-machine interface can be tailored to ISAS needs and to some target hardware, thus increasing its efficiency significantly. Other components of the software can also be made to run more efficiently with further effort. With some effort, core memory requirements could be reduced by swapping data in and out of an offline storage device such as a disk if this is necessary to fit the system into the target hardware. Graphical inputs, achieved through the use of a track ball, joy stick, mouse, or touch pad, would allow the user to specify start and end points, draw desired paths, or pose a query concerning a particular sensor platform by working directly with the image on the graphics monitor.

## 6.2 SUMMARY OF WORK PERFORMED

AI&DS successfully demonstrated that the dynamic programming mathematical technique can be used to solve the problem of planning naval transits that avoid enemy surveillance optimally. Also demonstrated was a friendly user environment that facilitates interaction between the user and the system, providing assistance when needed, so that a user does not require computer experience in order to use the system.

The dynamic programming algorithm and its support software were written to demonstrate that the technique could work for the surveillance avoidance problem. For a number of reasons, budgetary limitations being the most significant, engineering efficiency was not emphasized

strongly in developing the software. The resulting software demonstrates feasibility of concept, but further work is necessary to prove feasibility of implementing the system on a smaller Navy machine so that memory constraints can be satisfied without exceeding reasonable bounds on runtime. This further development would be guided by the target hardware, either aboard ship or shore-based, and by the required response time of the system, which would depend on whether the aid is to be used in real time or as an advance planning tool.

The problem treated by the DP software is a simplification of the actual surveillance avoidance problem. On the positive side, the simplified problem includes *most* of the components of the realistic surveillance avoidance problem, though some of the models are simple. The program does admit three platform types: satellites, fixed platforms on land or in the water, and various moving surface vessels. A number of simple sensor models are provided, the software can handle platforms with multiple sensors, and the modularity of the software allows for fairly straightforward substitution of realistic sensor models. A rendezvous point and a number of time-dependent avoidance areas are allowed. On the negative side, deterministic position and motion data are used for all platforms. Realistically, surface vessels should be modeled stochastically and random unknown threats should be included. More than one rendezvous point may be desirable. The effects of weather on motion and on sensor performance are currently ignored. The cost function of minimum cumulative probability of detection is not the function of interest in some cases.

The man-machine interface provides easy interactions between user and system. It relies on a collection of selection menus and data forms that the user interacts with through cursor control keys as well as with standard keyboard characters. The interface posts error messages to the user that aid the user in correcting input mistakes. It is also designed to provide explanations on the use of various ISAS capabilities on request. The new user should have little difficulty learning how to use the system, asking the system for assistance whenever needed, while the experienced user can move quickly through the forms. The

capabilities to generate and display graphical outputs and to provide printouts of results are also provided by the interface.

The current implementation of the interface is not without its disadvantages. At the heart of the interface is an experimental relational data base system called TROLL, and buffer subroutines to TROLL are currently written in LISP and C. This approach was taken to make use of an existing AI&DS program to develop a generic man-machine interface. The interface is slow because it is generic in nature rather than designed specifically for the needs of the surveillance avoidance project, the languages used are complex and somewhat slow, and the TROLL system is still in a developmental stage. Also, most Navy machines would not be capable of supporting this interface, so that a translation of the software into some other language would probably be necessary in order to transport the ISAS system. Candidate languages include C, ADA, and FORTRAN, depending on the capabilities of the target hardware.

Together, the problem solving software and the man-machine interface demonstrate that ISAS does have the potential to be a significant aid to the user interested in planning surveillance avoidance routes. ISAS can determine a minimum probability of detection path against a number of platforms of different types and with different sensor configurations, simultaneously satisfy constraints on fuel consumption, rendezvous requirements, and avoidance regions, and generate its results in far less time than a human analyst could. With further development to extend the capabilities of the system and to engineer an efficient system for some target hardware, ISAS could greatly enhance the Navy transit planning process.

## APPENDIX A. DYNAMIC PROGRAMMING

Dynamic programming (DP) is an optimization procedure that can be applied to a wide range of multistage decision problems. This Appendix begins with a heuristic explanation of the DP procedure. Section 2 gives the mathematical description of DP. In Section 3 the formulation of the surveillance avoidance problem as a DP problem is given.

### A.1 HEURISTIC EXPLANATION

Figure A-1(a) shows a one-dimensional forward dynamic programming problem. The stage represents a point at which a decision must be made; the state characterizes the state of the world at the point a decision is to be made. In this example, stage is time and state is position of a ship moving in one dimension. The problem is to find an optimal path that starts in any state at stage 1 and ends in any state at stage 5. For this example the performance measure or figure of merit (FOM) to be maximized for a path is the sum of the numbers in the cells that the path passes through. For example, if the ship stays in state 4 at every stage the total FOM would equal  $6+5+4+7+5=27$ . The arrows in stage 3, state 2 (3,2) of Fig. A-1(b) show the directions from which a ship in stage 2 could enter (3,2). It is assumed that the ship speed is such that a ship cannot jump more than one state in one stage (e.g., a transition from (2,4) to (3,2) is not feasible).

Figure A-1(c) shows the first stage of processing. For each state in stage 2 the optimal path from the first stage to the second has been determined. For example, for a path to (2,4) there are three possible origins: (1,5), (1,4), and (1,3). The two-stage FOM's for these are 13, 11, and 12, respectively. The first FOM is the maximum and it is written in the lower left-hand corner of (2,4), and the best direction (control vector) is noted by a vector pointing back to (1,5). The computation proceeds similarly for the other states in stage 2.

5	8	7	8	8	7
4	6	5	4	7	5
3	7	10	5	6	4
2	6	6	7	7	4
1	10	2	5	6	8
	1	2	3	4	5

STATE →

STAGE →

Figure A-1(a) PROBLEM

5	8	7	8	8	7
4	6	5	4	7	5
3	7	10	5	6	4
2	6	6	7	7	4
1	10	2	5	6	8
	1	2	3	4	5

STATE →

STAGE →

Figure A-1(b) FEASIBLE TRANSITIONS

5	8	7	8	8	7
4	6	5	4	7	5
3	7	10	5	6	4
2	6	6	7	7	4
1	10	2	5	6	8
	1	2	3	4	5

STATE →

STAGE →

Figure A-1(c) RESULTS AT 2nd STAGE

5	8	7	8	8	7
4	6	5	4	7	5
3	7	10	5	6	4
2	6	6	7	7	4
1	10	2	5	6	8
	1	2	3	4	5

STATE →

STAGE →

Figure A-1(d) FINAL RESULT

Figure A-1: Dynamic Programming Example

The second phase of processing determines the optimal transition from the first stage to the third stage for each state in the third stage. For example, the possible transitions into (3,3) are from (2,4), (2,3), or (2,2). The corresponding three-stage FOM's are  $13+5=18$ ,  $17+5=22$ , and  $16+5=21$ . The second is the optimum, and it is noted along with the best direction.

The computation proceeds sequentially to the last stage (Fig. A-1(d)). The best path can be obtained by scanning the FOM's in the last stage. State 1 has the highest FOM (39), and the optimal path is the sequence (1,3), (2,3), (3,2), (4,2), (5,1), obtained by following the control vectors backward.

The dynamic programming approach has many desirable properties, including a guarantee of global optimality. It is a recursive procedure that requires the results from stage N, but not the results from earlier stages, to process stage N+1. Once the first stage values are determined (this task is normally trivial) the remaining stages can be processed sequentially. The main disadvantage of DP is that computational and memory requirements can become unmanageable if the size of the state space or the number of stages is allowed to grow too large.

## A.2 MATHEMATICAL DESCRIPTION

In discrete dynamic programming, the stage (decision point) can be represented by an integer k. The state of the world at stage k is summarized by the vector  $x(k)$ . The decision made at stage k is represented by the vector  $u(k)$ . The state transition equation is

$$x(k+1) = f[x(k), u(k), k], \quad (1)$$

which can also be written for purposes of forward, rather than backward, DP as

$$x(k) = g[x(k+1), u(k), k]. \quad (2)$$

That is, given the state at stage k and the decision made at k-1, the state at k-1 can be determined. For forward DP a reverse of the

Markovian state property is required: The optimal sequence of decisions that leads from the initial state to the current state is independent of the decisions that will be made at future stages. This allows one to begin at the first stage and move forward, finding partial sequences of optimal decisions before the remaining stages have been considered.

The objective is to find the decision sequence that leads from the initial state to the final state and optimizes some figure of merit (FOM) function. The FOM must be a separable function, meaning that it must be possible to calculate the contribution of a decision to the FOM without needing to know any of the other decisions. The FOM is often a sum or product of some function of the state, control, and stage, for example

$$J = \sum_{k=1}^{K-1} h[x(k), u(k), k]. \quad (3)$$

Consider the problem of maximizing  $J$  in (3) over all possible sequences of  $u(k)$ . The solution procedure begins by finding the optimal policy for each state at the first stage, which is normally a trivial problem. In going from the initial state to the state  $x(k+1)$ , the optimal FOM is

$$\begin{aligned} \hat{J}_{k+1}[x(k+1)] &= \max_{u(1), \dots, u(k)} \left\{ \sum_{j=1}^k h[x(j), u(j), j] \right\} \quad (4) \\ &= \max_{u(1), \dots, u(k)} \{ h[x(k), u(k), k] + \sum_{j=1}^{k-1} h[x(j), u(j), j] \} \\ &= \max_{u(k)} \{ h[x(k), u(k), k] \} + \max_{u(1), \dots, u(k-1)} \left\{ \sum_{j=1}^{k-1} h[x(j), u(j), j] \right\}, \end{aligned}$$

where the last equality comes from the fact that  $h[x(k), u(k), k]$  is independent of controls from 1 to  $k-1$ . Recognizing the last term of (4) as the maximum FOM from stage 1 to  $k-1$ , (4) becomes

$$\hat{J}_{k+1}[x(k+1)] = \max_{u(k)} \{ h[x(k), u(k), k] + J_k[x(k)] \}. \quad (5)$$

From (5), the control  $u(k)$  at stage  $k$  that maximizes the FOM from the first stage to stage  $k$  is that which maximizes  $h$  when using control  $u(k)$  from state  $x(k)$  plus the sum of the  $h$ 's when proceeding optimally to  $x(k)$  from the initial state  $x(1)$ , the latter quantity having been calculated during the previous stage.

Using this recursive relationship, the solution procedure moves forward, stage by stage, each time finding the optimal policy for each state of that stage, until it finds the optimum policy that ends at the final state.

### A.3 DP FOR SURVEILLANCE AVOIDANCE

To formulate the surveillance avoidance problem as a DP problem, let the stage be an increment of a fixed decision interval. That is, a new decision must be made every  $\Delta t$  units of time, where  $\Delta t$  is a constant. The state is the position of the evading ship at any given stage.

Position is quantized as a two-dimensional square grid. For each cell at each stage, the decision to be made is to choose a transition to a new cell that is reachable from the current cell within the  $\Delta t$  unit decision interval. The number of transitions to be considered from any cell depends on the resolution option chosen by the user. Transitions that pass through land masses or exclusion areas are discarded. The FOM of a transition is the probability of detection as the evading ship makes that transition at that time. This probability of detection of a transition depends on the relative positions of the sensors to the position of the evader.

Consider the situation where there are  $N$  sensors with detection capability described as follows. If a platform is located at  $x$  in a two-dimensional space in the time interval  $[t, t+\Delta t)$ , then the  $i^{\text{th}}$  sensor can detect it with probability  $P_D^i(x, t)$ . If the detection capabilities of all the sensors are independent from each other, then the probability that the platform, when located at  $x$  in  $[t, t+\Delta t)$ , is not detected by any one of the sensors is given by

$$P_{ND}(x, t) = \prod_{i=1}^N (1 - P_D^i(x, t)) \quad (6)$$



This can be visualized via Fig. A-2. The two-dimensional surface is divided into grids as shown in the figure. Assume that  $N=3$ . At a particular time interval  $[t, t+\Delta t)$ , the coverage region of sensor 1 ( $P_D(x, t) > 0$ ) does not overlap with the coverage regions of sensors 2 and 3, whereas sensors two and three have an overlapping region. A "value" that represents the probability of nondetection given by (6) is put into each cell. For example, a cell located at  $x$  which is not covered by any one of the sensors contains a value of 1, representing nondetection probability equal to 1; cells which are covered by only the  $i^{\text{th}}$  sensor have entries  $1-P_D^i(x, t)$ ,  $i=1, 2, 3$ ; and cells which are in the overlapping regions of sensors 2 and 3 have entries  $(1-P_D^2(x, t))(1-P_D^3(x, t))$ .

Due to sensor motion, the value in each cell can change as time evolves. This can be represented by a three-dimensional space with time in the third coordinate (see Fig. A-3). The three-dimensional space can be divided into cells with non-detection probability entered into each cell as was discussed above.

A mission of traveling from an initial position to a final destination at a specific time can be represented by requiring that the platform starts from a particular cell at time  $k=0$  (i.e., lowest layer in the three-dimensional grid in Fig. A-3), and arrives at a particular cell at the top layer at time  $k=K$ . The surveillance avoidance problem is to achieve the given mission with minimum probability of detection while satisfying the fuel constraint. This can be formulated as follows: Given  $x(0)=x_0$  (in the lowest layer),  $x(K)=x_T$  (in the top layer), find a path from  $x_0$  to  $x_T$  such that the nondetection probability along the path  $X$  (originating from  $x_0$  and ending at  $x_T$ )

$$P_{ND}(X) = \prod_{k=1}^K \prod_{i=1}^N [1-P_D^i(x(k\Delta t), k\Delta t)] \quad (7)$$

is maximized subject to the fuel constraint.

This is a separable FOM that fits nicely into the framework of dynamic programming. The fuel constraint could be handled by augmenting

DIVIDE  $x_1, x_2$  INTO GRID

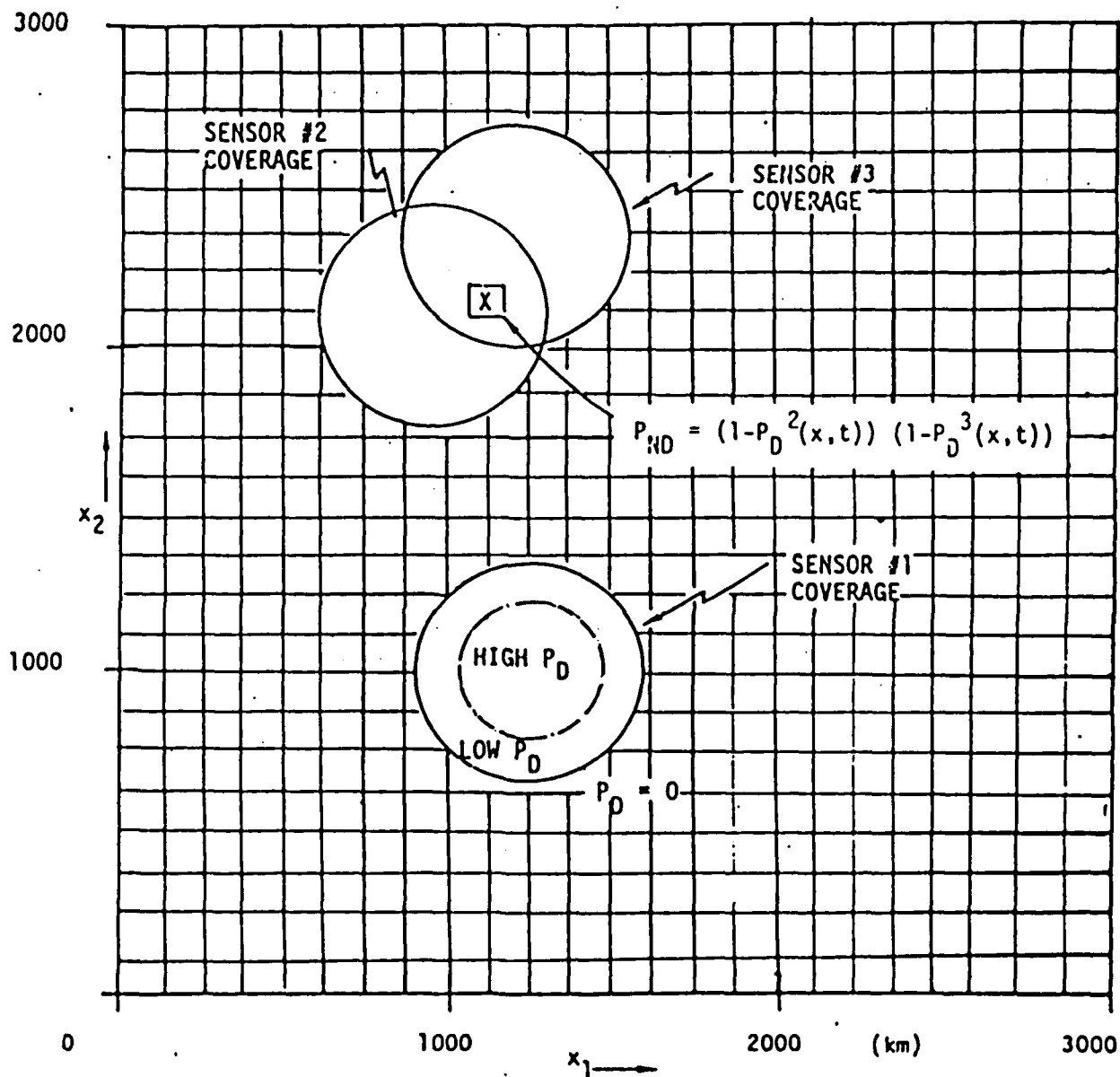


Figure A-2: Probability of Non-Detection Grid (time = t)

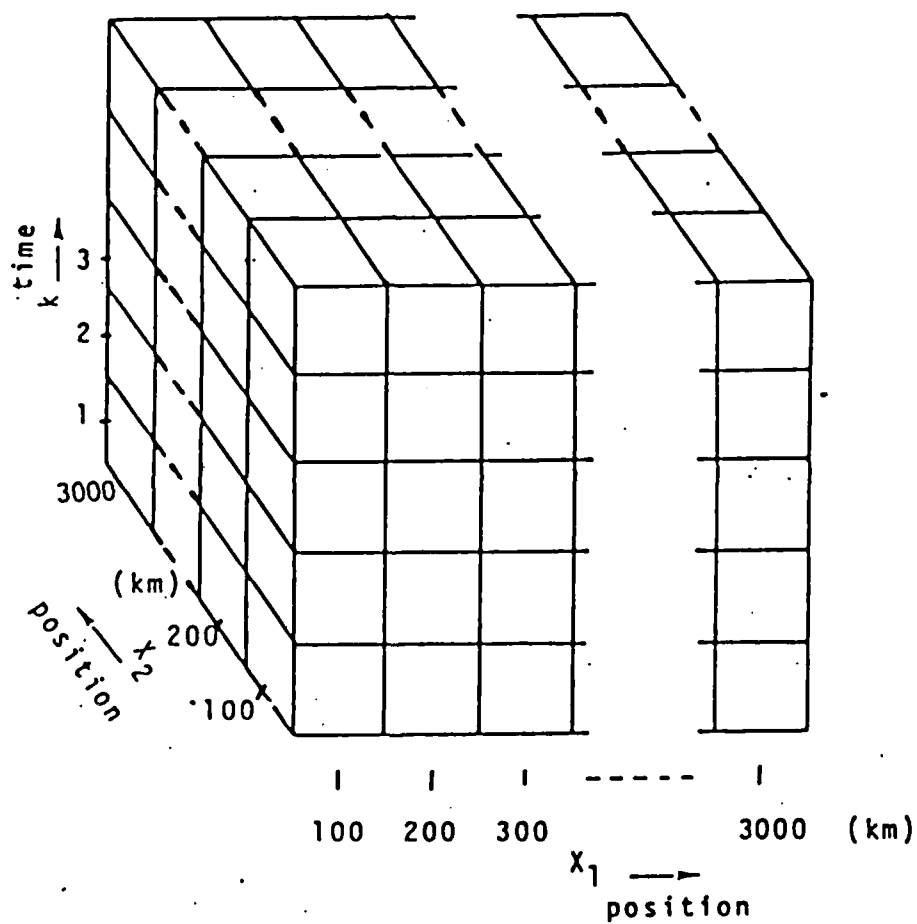


Figure A-3: Three-Dimensional Grid Space

the state variable to include net fuel consumption in addition to position. However, it is computationally more efficient for this problem to incorporate the fuel constraint into the FOM by way of the Lagrange multiplier technique. This converts the constrained problem to an unconstrained one.